IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF TEXAS
DALLAS DIVISION

GLOBAL TEL*LINK CORPORATION, *Plaintiff*, v. SECURUS TECHNOLOGIES, INC. *Defendant*.	§ § § § § § § § § §	CIVIL ACTION NO. 3:14-CV-00829-K **ECF**

APPENDIX TO PLAINTIFF GLOBAL TEL*LINK CORPORATION'S
SUPPLEMENTAL CLAIM CONSTRUCTION BRIEF

Plaintiff Global Tel*Link Corporation hereby submits this Appendix to Plaintiff Global Tel*Link Corporation's Supplemental Claim Construction Brief.

Description	**Appendix Number**
Steven M. Kaplan, *Wiley Electrical and Electronics Engineering Dictionary* (John Wiley & Sons, Inc. 2004)	App. 031-034
PR Newswire, *T-Netix Announces Fiscal 1998 Results of $0.07 vs. $0.06 Earnings Per Share,* PR Newswire Association, Inc. (Oct. 29, 1998)	App. 035-038
Business Wire, *T-Netix, Inc. Purchases Prepaid Calling Assets,* Business Wire, Inc. (July 3, 2002)	App. 039-040
U.S. Patent No. 6,836,540 (filed July 3, 2002) (issued Dec. 28, 2004)	App. 041-057
U.S. Patent No. 5,974,114 (filed Sept. 25, 1997) (issued Oct. 26, 1999)	App. 058-082
NCIC News Release, *NCIC introduces new Prepaid Calling Card Program for wholesale and retail services,* NCIC Inmate Phone Service (Sept. 1, 1999)	App. 083-086

Dated: July 20, 2015

Respectfully submitted,

/s/ J.C. Rozendaal
J.C. Rozendaal (*pro hac vice*)
Courtney S. Elwood (*pro hac vice*)
Evan T. Leo (*pro hac vice*)
Kenneth M. Fetterman (*pro hac vice*)
Christopher C. Funk (*pro hac vice*)
Nicholas O. Hunter (*pro hac vice*)
KELLOGG, HUBER, HANSEN, TODD,
EVANS & FIGEL, P.L.L.C.
1615 M Street, N.W., Suite 400
Washington, DC 20036
Tel: (202) 326-7900
Fax: (202) 326-7999

E. Leon Carter
Texas State Bar No. 03914300
lcarter@carterscholer.com
John Steven Torkelson
Texas State Bar No. 00795154
jtorkelson@carterscholer.com
Linda R. Stahl
Texas State Bar No. 00798525
lstahl@carterscholer.com
Sheria D. Smith
Texas State Bar No. 24075097
ssmith@carterscholer.com
CARTER SCHOLER ARNETT
HAMADA & MOCKLER PLLC
8150 N Central Expressway
Fifth Floor
Dallas, TX 75206
Tel: (214) 550-8188
Fax: (214) 550-8185

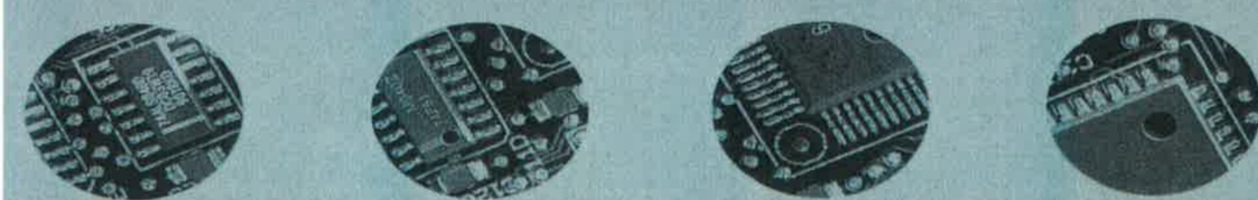
*Counsel for Plaintiff Global Tel*Link Corporation*

CERTIFICATE OF SERVICE

I hereby certify that on July 20, 2015, Plaintiff electronically filed the foregoing document with the Clerk of the Court, using the CM/ECF system, which will send certification of such filing to all counsel of record.

/s/ J.C. Rozendaal

WILEY ELECTRICAL AND ELECTRONICS ENGINEERING DICTIONARY



STEVEN M. KAPLAN

WILEY ELECTRICAL AND ELECTRONICS ENGINEERING DICTIONARY

Steven M. Kaplan
Lexicographer

IEEE
IEEE PRESS

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

For general information on our other products and services please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993 or fax 317-572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic format.

Library of Congress Cataloging-in-Publication Data is available.

Kaplan, Steven M.

Wiley Electrical and Electronics Engineering Dictionary

ISBN 978-0-471-40224-4

ıstic-leadless chip carrier A plastic chip package which is ıermetically sealed, and which instead of leads consisting of netal prongs or wires, has metallic contacts called castellaions which are flush with the package or recessed. The contacts are usually on all four sides of the package, alhough they may be located elsewhere. Its abbreviation is PLCC.

stic package A chip package made of plastic. Also called **plastic chip package**.

stic PGA Abbreviation of **plastic pin grid array**.

stic pin grid array A plastic package capable of providing ıp to several hundred pins, all located on its underside. Jsed, for instance, to package computer chips. The design seeks to minimize the distance signals must travel from the chip to each designated pin. Its abbreviation is **PPGA**, or **plastic PGA**.

stic quad flat-pack A plastic surface-mount chip package n the form of a square, which provides leads on all four sides. Such a package affords a high lead count in a small ırea. Its abbreviation is **PQFP**. Also spelled **plastic quad latpack**. Also called **plastic quad flat-package**.

stic quad flat-package Same as **plastic quad flat-pack**.

stic quad flatpack Same as **plastic quad flat-pack**.

te **1.** The positive electrode in an electron tube. Electrons emitted by the cathode travel towards it. Also known as **anode (1)**. **2.** One of the electrodes in a capacitor. Also called **capacitor plate**. **3.** One of the electrodes in an electrochemical cell. **4.** The formation of a metal deposit on another surface, usually a different metal, via electrolysis. Also called **electroplate**. **5.** A thin, or very thin, layer or coat deposited or applied to a metal. Also, to apply such a plate. **6.** A light-sensitive sheet, usually of glass or metal, upon which a photographic image may be recorded.

te battery A battery which supplies the anode, or plate, current in electron tubes. Also known as **anode battery**, or **B battery (1)**.

te capacitance In an electron tube, the capacitance between the plate and another electrode, especially the cathode. Also called **anode capacitance**.

te characteristic In an electron tube, the variation of the current of the anode relative to the voltage applied to it. Also known as **anode characteristic**.

te circuit A circuit which includes all of the components connected between the anode and the cathode in an electron tube, including the anode voltage source. Also called **anode circuit**.

te current In an electron tube, the electron flow from the cathode to the anode. Also known as **anode current**.

te detection The function of a **plate detector**.

te detector An electron tube detector in which the anode circuit rectifies the input signals. Also called **anode detector**.

te dissipation In an electron tube, the power dissipated by the anode in the form of heat. This loss is caused by the anode being bombarded by electrons and anions. Also called **anode dissipation**.

te efficiency In an electron tube, the ratio between the AC load circuit power and the DC anode input power. Also called **anode efficiency**.

te-grid capacitance In an electron tube, capacitance between the plate and the control grid.

te impedance In an electron tube, the total impedance between the anode and the cathode, without taking into account the electron stream. Also called **plate-load impedance**, **anode-load impedance**, or **anode impedance**.

te input power In a vacuum tube, the DC power consumed by the anode. Also called **plate power input**, or **anode input power**.

plate-load impedance Same as **plate impedance**.

plate modulation In an electron tube, amplitude modulation obtained by varying the voltage of the anode proportionally to the fluctuations in the modulating wave. Also called **anode modulation**.

plate potential Same as **plate voltage**.

plate power input Same as **plate input power**.

plate power supply In an electron tube, the DC applied to the anode to place it at a high current potential relative to the cathode. Also called **plate supply**, or **anode supply**.

plate pulse modulation In an electron tube, modulation produced by applying external voltage pulses to the anode. Also called **anode pulse modulation**.

plate resistance In an electron tube, the ratio of a minimal change in the anode voltage to a minimal change in the anode current. All other voltages must be held constant. Also called **anode resistance**.

plate saturation In an electron tube, the condition in which the plate current can not be further increased, regardless of any additional voltage applied to it, since essentially all available electrons are already being drawn to said plate. Also called **anode saturation**, **current saturation**, **voltage saturation**, or **saturation (3)**.

plate spacing In a capacitor, the distance between plates. The capacitance value of a capacitor varies as a function of various variables, including the distance between its plates. A variable capacitor, for instance, may utilize plates which are moved relative to each other.

plate supply Same as **plate power supply**.

plate voltage Also called **plate potential**, or **anode voltage**. **1.** In an electron tube, the difference in potential between the anode and the cathode. **2.** In an electron tube, the difference in potential between the anode and a specific point of the cathode.

plateau Within the characteristic curve of a component or device, such as a tube or transistor, a region where an increase in one variable, such as current, has little or no effect on another variable, such as voltage. Also, any such flat region within a response curve.

plated-through hole Within a printed-circuit board, a hole made through plating, and which serves to mount and connect components by connecting their pins or leads. Its abbreviation is **PTH**. Also called **pin-through hole**.

platen A cylindrical roller against which a print head strikes in an impact printer, typewriter, or similar device. A paper, or other suitable medium which is being printed upon, is guided and supported by this roller.

platform A specific hardware and software configuration, including the operating system. When a program or hardware device will only function properly with a particular platform, it is called **platform-dependent**, while those which can work across multiple platforms are called **platform-independent**. Also called **computing platform**.

platform-dependent Computer software or hardware which is designed to work properly only with a specific platform. Assembly language, for instance, is platform-dependent. This contrasts with **platform-independent**, in which software or hardware can work with more than one type of platform. Also called **architecture-dependent**.

Platform for Internet Content Selection Its abbreviation is **PICS**. A rating system utilized to classify Web sites. PICS itself provide no ratings, but promotes a uniform template for others to utilize.

platform-independent Computer software or hardware which is designed to work properly with more than one type of platform. An interpreter version of LISP is platform-independent, as is Java. This contrasts with **platform-dependent**, in which software or hardware is designed to

41 of 44 DOCUMENTS

PR Newswire

October 29, 1998, Thursday

SECTION: Financial News

DISTRIBUTION: TO BUSINESS EDITOR

LENGTH: 1804 words

HEADLINE: T-NETIX Announces Fiscal 1998 Results of $0.07 vs. $0.06 Earnings Per Share**.**

DATELINE: ENGLEWOOD, Colo., Oct. 29

BODY:

T-NETIX, Inc. (Nasdaq: TNTX), a leading provider of specialized call processing and fraud control software technologies announced today results for the fiscal year ended July 31, 1998.

Revenue increased 5% to $38,213,000 for the year ended July 31, 1998 from $36,292,000 a year ago. Net earnings were $599,000 or $0.07 per common share (diluted) for the year compared to $591,000 or $0.06 per common share (diluted) in fiscal 1997. The revenue increase in fiscal 1998 resulted primarily from the net addition of call processing systems for both existing and new customers and the increase of direct call provisioning revenue.
The Company processed 93.3 million calls for the year ended July 31, 1998, an increase of 3% over the 90.9 million calls processed for the same period a year ago. There was a 4% increase in telecommunications services revenue to $34,860,000 for the year ended July 31, 1998 compared to $33,471,000 for the same period a year ago. In addition, there was an 80% increase in direct revenue to $2,516,000 for the year ended July 31, 1998 compared to $1,395,000 for the same period a year ago. Exclusive of the one-time licensing fee of $750,000 for the year ended July 31, 1997, revenue and EBIT for the year ended July 31, 1998 increased 8% and 54%, respectively, compared to the same period a year ago.

Gross margin decreased 2% to $19,904,000 for the year ended July 31, 1998 from $20,347,000 for the same period a year ago. Excluding the one-time license revenue the gross margin would have increased 2% for the year ended July 31, 1998. The gross margin decrease was due primarily to increases in telecommunications services and direct call provisioning expenses. The gross margin decrease was offset by reductions in selling, general and administrative expenses and research and development expenses. EBIT decreased by 5% to $1,852,000 for the year ended July 31, 1998 from $1,955,000 for the same period a year ago primarily due to reduction in gross margin. EBITDA (earnings before interest, taxes, depreciation and amortization) increased to $10,102,000 for the year ended July 31, 1998 from $10,090,000 a year ago. However, EBITDA for the ICS Division increased to $13,174,000 for the year ended July 31, 1998 from $12,049,000 a year ago, or a 6% increase.

"The ICS Division continues to provide earnings growth for the Company despite somewhat slower revenue growth," said Al Schopp, President of Inmate Calling Services. "If you adjust last year's EBIT for the $750,000 one-time license fee we received in fiscal 1997, the ICS Division increased EBIT by 51% to $5,848,000 for the year ended July 31, 1998 from $3,869,000 last year. It is also significant that we were able to improve our financial position while making considerable investment in the research and development of our new inmate calling platform," Schopp added.

"We are pleased to see the overall results for the current year, particularly when considered in light of last year's one-time license fee," said Tom Huzjak, T-NETIX Chief Executive Officer and Chairman of the Board. "We have managed to maintain a steadily improving position in our core business, the ICS Division, while continuing to invest in the opening of new markets with our SpeakEZ Division," Huzjak added. "During the year, we have made considerable progress towards the achievement of our goal of creating and capitalizing on the emerging voice verification market. Through our marketing efforts, there now is awareness in our target markets -- call centers, telecommunications, computer/network security and government services -- of the availability and benefits of speaker verification technology. Our current installations and strategic market pilots with key industry leaders are expected to further the creation of a 'pull' factor for our SpeakEZ Voice Print(SM) products. We have made considerable progress in the development of distribution alliances with the major providers of complementary technology in our target markets to be able to meet prospective customer demand for deliverable, integratable and scalable solutions. Our newest distribution partner, Nissho Iwai Infocom Systems, Ltd., a division of one of the largest Japanese conglomerates, is illustrative of the scope and quality of our growing international distribution channel. Overall, we are very pleased with both the financial and developmental progress achieved over this last year."

Based in Englewood, Colorado, T-NETIX provides specialized call processing services on a transaction fee basis to the telecommunications industry including AT&T, Bell Atlantic, US WEST, SBC Communications, Inc., GTE and other call providers. The Company's patented Strike Three(TM) , three-way call prevention software and its proprietary call processing technologies prevent telecommunications fraud. T-NETIX offers its SpeakEZ Voice Print(SM) speaker verification technology as a fraud prevention tool to the telecommunications, financial services, computer network and industries. SpeakEZ Voice Print(SM) is based on patented (U.S. Patent 5,522,012) and patent pending technologies.

T-NETIX notes that actual results may differ from the forward-looking statements made above which involve risks and uncertainties discussed more completely in the T-NETIX, Inc. Annual Report on Form 10-K and the T-NETIX, Inc. Quarterly Reports on Form 10-Q.

T-NETIX, INC.
FINANCIAL HIGHLIGHTS
(In thousands except for per share data)

	Years Ended July 31, 1998	1997
REVENUE:		
Telecommunications services	$34,860	33,471
Telecommunications licensing	205	750
Direct call provisioning	2,516	1,395
Voice print	632	676
Total revenue	38,213	36,292
EXPENSES:		
Operating costs and expenses:		

Telecommunications services	15,920	14,478
Direct call provisioning	2,311	1,297
Voice print	78	170
Total operating costs and expenses	18,309	15,945
Selling, general and administrative	7,448	7,488
Research and development	2,354	2,769
Depreciation and amortization	8,250	8,135
Total expenses	36,361	34,337
Operating income	1,852	1,955
Interest expense	(1,055)	(932)
Earnings before income taxes	797	1,023
Income taxes	(198)	(432)
Net earnings	$599	$591
Basic earnings per common share	$0.07	$0.06
Diluted earnings per common share	$0.07	$0.06
Weighted average common shares - basic	8,493	8,232
Weighted average common shares - diluted	9,206	9,156

Segment Information

The Company operates in two business segments; the ICS Division and the SpeakEZ Division. Segment information for the years ended July 31, 1998 and 1997 are as follows:

Year Ended July 31, 1998
(amounts in thousands)

	Inmate Calling Services	SpeakEZ	Total
Revenue from external customers	$37,581	$632	$38,213
Operating income (loss)	5,848	(3,996)	1,852
Depreciation and amortization	7,326	924	8,250
Interest expense	463	592	1,055
Segment earnings (loss) before taxes	5,385	(4,588)	797
Segment assets	52,349	4,833	57,182

Year Ended July 31, 1997
(amounts in thousands)

	Inmate Calling Services	SpeakEZ	Total
Revenue from external customers	$35,616	$676	$36,292
Operating income (loss)	4,619	(2,664)	1,955
Depreciation and amortization	7,430	705	8,135
Interest expense	565	367	932
Segment earnings (loss) before taxes	4,054	(3,031)	1,023
Segment assets	50,432	5,544	55,976

Note: News releases and other information on T-NETIX can be accessed at no charge at www.ctaonline.com/ir/tnetix.htm or www.T-NETIX.com on the Internet.

SOURCE T-NETIX, Inc.

CONTACT: Alvyn A. Schopp, Chief Financial Officer of T-NETIX, Inc., 303-790-9111; or Wayne Brown, Sr. Vice President, or Michele Hartley, Asst. Acct. Exec., of Carl Thompson Associates, 800-959-9677

LOAD-DATE: October 30, 1998

LexisNexis®

9 of 15 DOCUMENTS

Business Wire

July 3, 2002, Wednesday

DISTRIBUTION: Business Editors

LENGTH: 501 words

HEADLINE: T-NETIX, Inc. Purchases Prepaid Calling Assets

DATELINE: DALLAS, July 3, 2002

BODY:

T-NETIX, Inc. (Nasdaq:TNTX), announced that it has recently purchased from ACT Telecom, Inc., a Houston based prepaid calling platform provider and a wholly-owned subsidiary of ClearMediaOne, Inc., assets including the ACT telecommunications switch, prepaid calling platform and associated software.

Over the next three months the company expects to integrate the ACT prepaid services with the T-NETIX existing prepaid calling solution for the inmate communications market, resulting in a single robust prepaid platform. This allows T-NETIX to offer a more complete and cost-effective payment solution to its partners, facilities and customers.

"The purchase and integration of the ACT prepaid platform is yet another step in strengthening our core service offering to the inmate communications market, while at the same time allowing us the flexibility to broaden our reach to new markets," said Richard Cree, EVP Business Development for T-NETIX. "Given the current buyers' market and the resulting consolidation efforts going on in our industry, we believe this is the right time to engage in these type of asset acquisitions."

About T-NETIX, Inc.

T-NETIX is a leading provider of specialized telecommunications products and services, including security enhanced call processing, call validation and billing for the corrections communications marketplace. The Company provides its products and services to more than 1,400 private, local, county and state correctional facilities throughout the United States and Canada. We deliver these services through direct contracts with correctional facilities, and through contracts with some of the world's leading telecommunications service providers, including Verizon, AT&T, SBC Communications, Qwest and Sprint. For additional news and information, visit the company's Web site at www.T-NETIX.com.

Forward Looking Statements

Except for historical information contained herein, the statements in this release are forward-looking and made

pursuant to the safe harbor provisions of the Private Securities Litigation Reform Act of 1995. Forward-looking statements involve known and unknown risks and uncertainties, which may cause the company's actual results in future periods to differ materially from forecasted or expected results. Those risks include, among other things, the competitive environment in the industry in general and in the company's specific market areas, inflation, changes in costs of goods and services and economic conditions in general and in the company's specific market area. Those and other risks are more fully described in the company's filings with the Securities and Exchange Commission.

CONTACT: T-NETIX, Inc.
President & CEO
Tom Larkin, 972/241-1535
tom.larkin@t-netix.com
or
Chief Financial Officer
Hank Schopfer, 972/241-1535
hank.schopfer@t-netix.com
or
Corporate Communications
Christine Stoneking, 949/858-2006
christine.stoneking@t-netix.com

URL: http://www.businesswire.com

LOAD-DATE: July 4, 2002

US006836540B2

(12) **United States Patent**
Falcone et al.

(10) **Patent No.: US 6,836,540 B2**
(45) **Date of Patent: Dec. 28, 2004**

(54) **SYSTEMS AND METHODS FOR OFFERING A SERVICE TO A PARTY ASSOCIATED WITH A BLOCKED CALL**

(75) Inventors: **Richard Falcone**, Addison, TX (US); **Keith S. Kelson**, Dallas, TX (US); **Jeremy W. Duke**, Fort Worth, TX (US); **Lee R. Johnson**, Plano, TX (US); **Robert E. Sullivan**, Plano, TX (US); **Randy Hoffman**, Plano, TX (US)

(73) Assignee: **Evercom Systems, Inc.**, Irving, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 137 days.

(21) Appl. No.: **10/190,315**

(22) Filed: **Jul. 3, 2002**

(65) **Prior Publication Data**

US 2003/0086546 A1 May 8, 2003

Related U.S. Application Data

(63) Continuation-in-part of application No. 10/135,883, filed on Apr. 29, 2002.

(51) **Int. Cl.**[7] **H04M 15/00**; H04M 17/00

(52) **U.S. Cl.** **379/127.02**; 379/121.04; 379/144.02; 379/184

(58) **Field of Search** 379/111, 121.04, 379/114.04, 114.14, 126, 114.2, 121.02, 145, 114.05, 115.01, 127.02, 144.02, 183, 184, 189, 191, 194, 201.01

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,797,910	A	*	1/1989	Daudelin	379/88.01
5,185,781	A	*	2/1993	Dowden et al.	379/88.04
5,210,789	A	*	5/1993	Jeffus et al.	379/127.01
5,517,555	A	*	5/1996	Amadon et al.	455/408
5,615,408	A	*	3/1997	Johnson et al.	455/405
5,627,887	A	*	5/1997	Freedman	379/114.21
5,655,013	A	*	8/1997	Gainsboro	379/188
5,832,068	A	*	11/1998	Smith	379/114.14
5,937,044	A		8/1999	Kim	
5,963,625	A	*	10/1999	Kawecki et al.	379/127.01
6,282,276	B1		8/2001	Felger	
6,353,663	B1		3/2002	Stevens et al.	
6,377,938	B1		4/2002	Block et al.	
6,397,055	B1		5/2002	McHenry et al.	
6,405,028	B1		6/2002	DePaola et al.	
6,430,274	B1		8/2002	Winstead et al.	
6,434,378	B1		8/2002	Fougnies	
6,483,910	B1		11/2002	Council	
6,639,978	B2	*	10/2003	Draizin et al.	379/114.21
2001/0028705	A1		10/2001	Adams et al.	
2002/0025028	A1		2/2002	Manto	
2002/0106065	A1		8/2002	Joyce et al.	
2002/0115424	A1		8/2002	Bagoren et al.	
2002/0136374	A1		9/2002	Fleischer, III et al.	
2002/0147002	A1		10/2002	Trop et al.	
2003/0002639	A1	*	1/2003	Huie	379/114.27
2003/0008634	A1		1/2003	Laybourn et al.	
2003/0138084	A1	*	7/2003	Lynam et al.	379/114.14
2003/0162526	A1	*	8/2003	Ogman et al.	455/406
2003/0200182	A1	*	10/2003	Truitt et al.	705/75

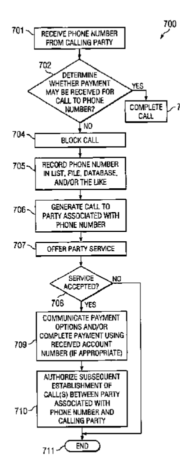
* cited by examiner

Primary Examiner—Binh Tieu
(74) *Attorney, Agent, or Firm*—Fulbright & Jaworski L.L.P.

(57) **ABSTRACT**

In one embodiment, the present invention is directed to a method of offering a service to a user. The method comprises receiving a dialed number from a request to initiate a collect telephone call from an origination source; processing the dialed number to determine whether payment may be received for the collect telephone call, wherein the processing blocks the collect call and stores the dialed number when the processing determines that payment may not be received for the collect telephone call; and establishing a telephone connection with the dialed number to communicate an offer for the service to a user associated with the dialed number.

47 Claims, 6 Drawing Sheets



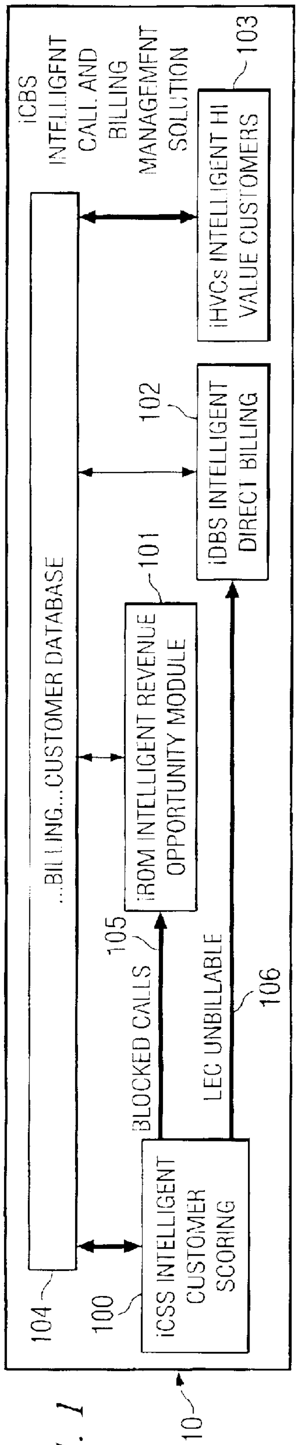

iCBS
INTELLIGENT CALL AND BILLING MANAGEMENT SOLUTION
...BILLING...CUSTOMER DATABASE
104
iCSS INTELLIGENT CUSTOMER SCORING
100
BLOCKED CALLS
105
LEC UNBILLABLE
106
iROM INTELLIGENT REVENUE OPPORTUNITY MODULE
101
iDBS INTELLIGENT DIRECT BILLING
102
iHVCs INTELLIGENT HI VALUE CUSTOMERS
103
10

FIG. 1

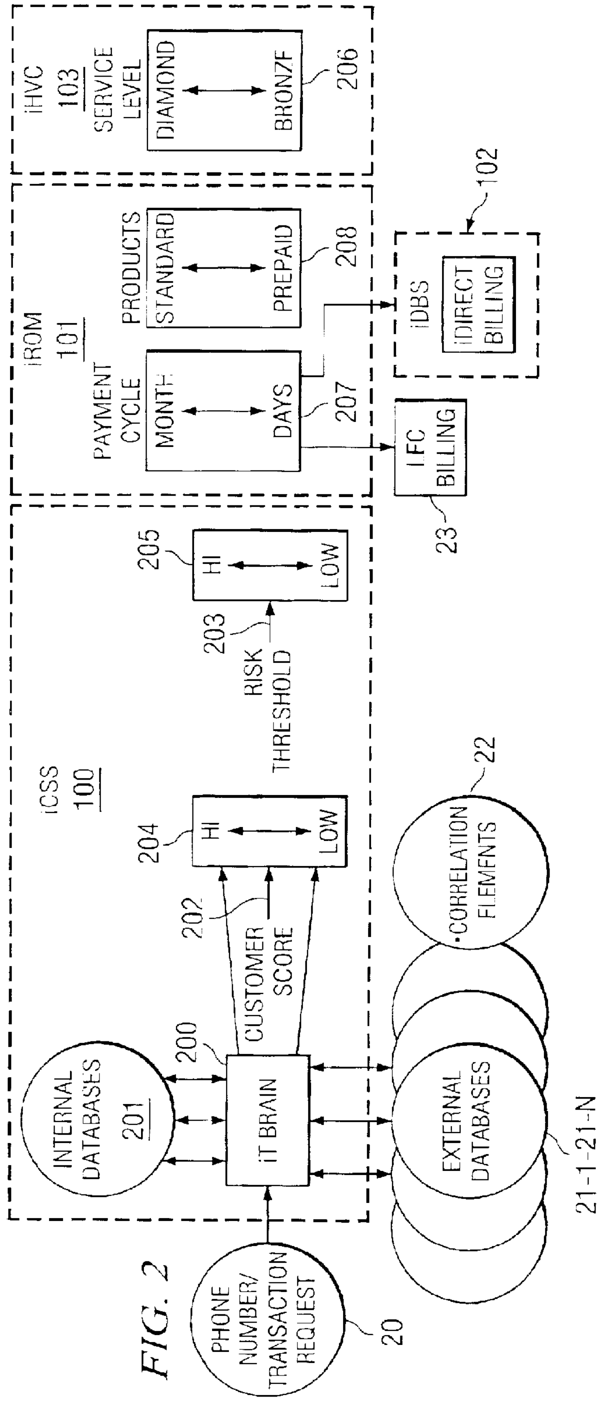

PHONE NUMBER/ TRANSACTION REQUEST
20
iCSS
100
INTERNAL DATABASES
201
iT BRAIN
200
CUSTOMER SCORE
202
HI
LOW
204
RISK THRESHOLD
203
HI
LOW
205
EXTERNAL DATABASES
21-1-21-N
CORRELATION ELEMENTS
22
iROM
101
PAYMENT CYCLE
MONTH
DAYS
207
PRODUCTS
STANDARD
PREPAID
208
iHVC
103
SERVICE LEVEL
DIAMOND
BRONZE
206
LEC BILLING
23
iDBS
iDIRECT BILLING
102

FIG. 2

FIG. 3

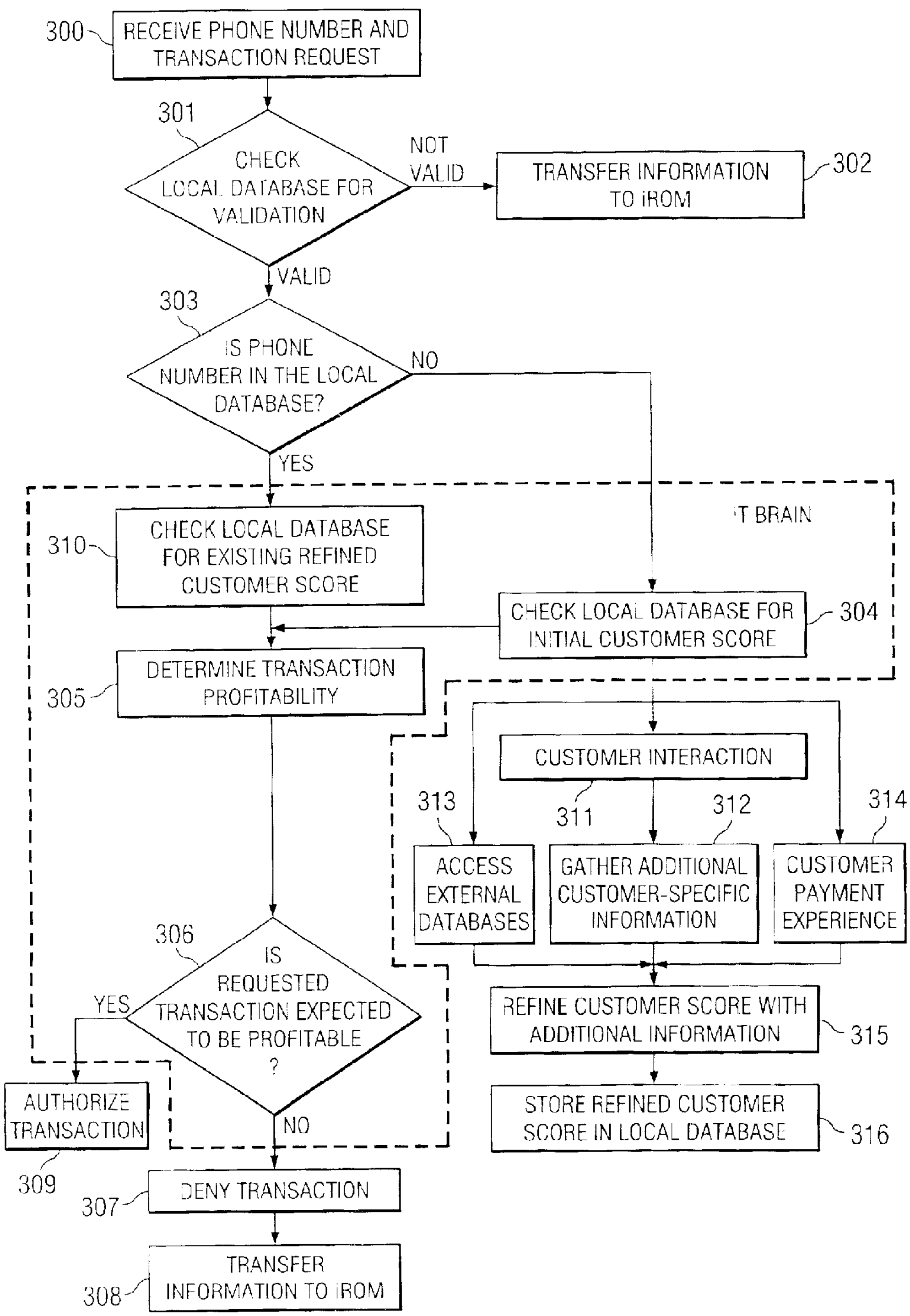
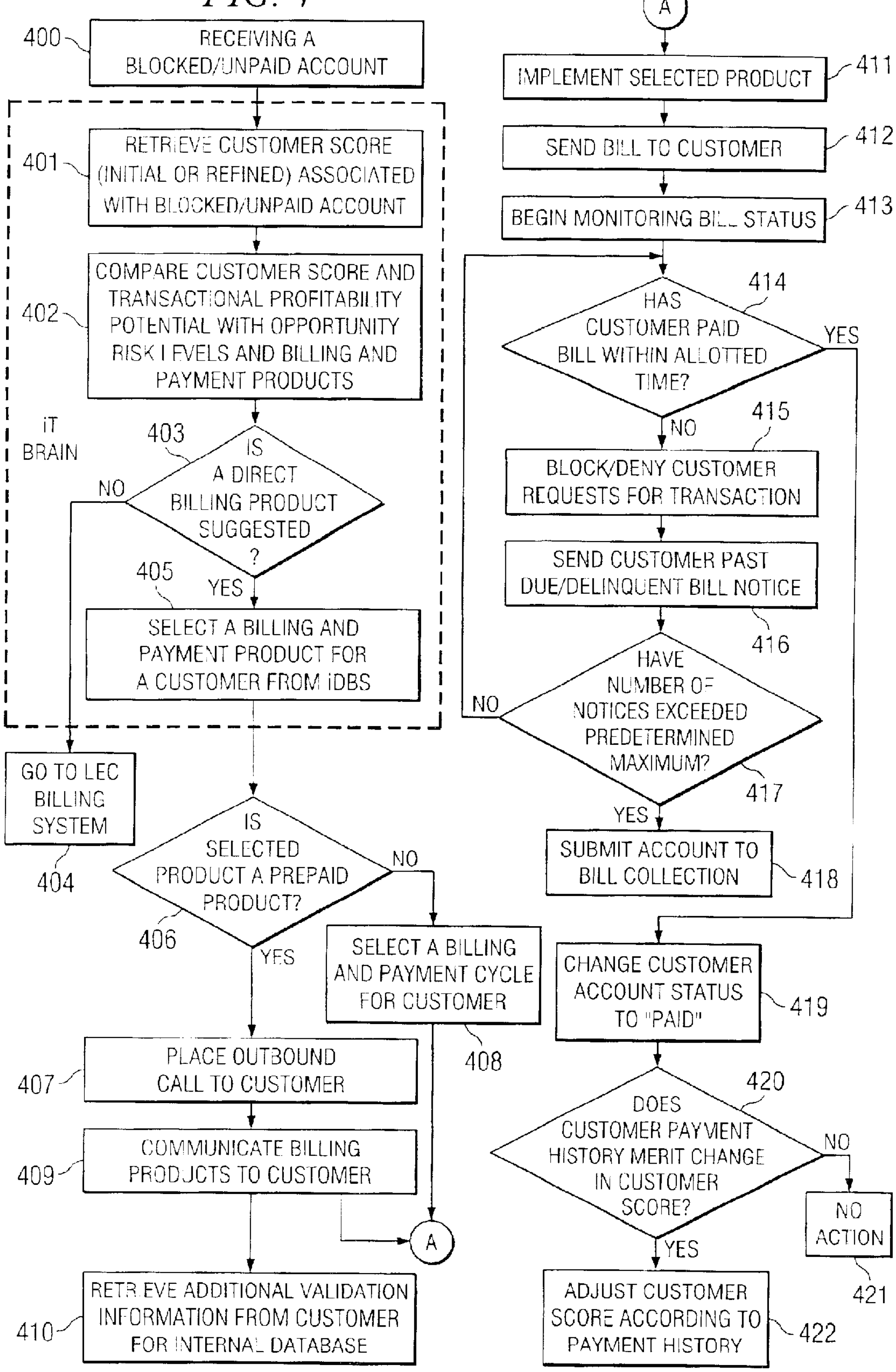

300
RECEIVE PHONE NUMBER AND TRANSACTION REQUEST
301
CHECK LOCAL DATABASE FOR VALIDATION
NOT VALID
TRANSFER INFORMATION TO iROM
302
VALID
303
IS PHONE NUMBER IN THE LOCAL DATABASE?
NO
YES
'T BRAIN
310
CHECK LOCAL DATABASE FOR EXISTING REFINED CUSTOMER SCORE
CHECK LOCAL DATABASE FOR INITIAL CUSTOMER SCORE
304
305
DETERMINE TRANSACTION PROFITABILITY
CUSTOMER INTERACTION
313
311
312
314
ACCESS EXTERNAL DATABASES
GATHER ADDITIONAL CUSTOMER-SPECIFIC INFORMATION
CUSTOMER PAYMENT EXPERIENCE
306
IS REQUESTED TRANSACTION EXPECTED TO BE PROFITABLE ?
YES
NO
REFINE CUSTOMER SCORE WITH ADDITIONAL INFORMATION
315
STORE REFINED CUSTOMER SCORE IN LOCAL DATABASE
316
AUTHORIZE TRANSACTION
309
307
DENY TRANSACTION
308
TRANSFER INFORMATION TO iROM

FIG. 4



400 RECEIVING A BLOCKED/UNPAID ACCOUNT
401 RETRIEVE CUSTOMER SCORE (INITIAL OR REFINED) ASSOCIATED WITH BLOCKED/UNPAID ACCOUNT
402 COMPARE CUSTOMER SCORE AND TRANSACTIONAL PROFITABILITY POTENTIAL WITH OPPORTUNITY RISK LEVELS AND BILLING AND PAYMENT PRODUCTS
iT BRAIN
403 IS A DIRECT BILLING PRODUCT SUGGESTED ?
NO
YES
405 SELECT A BILLING AND PAYMENT PRODUCT FOR A CUSTOMER FROM iDBS
404 GO TO LEC BILLING SYSTEM
406 IS SELECTED PRODUCT A PREPAID PRODUCT?
NO
YES
408 SELECT A BILLING AND PAYMENT CYCLE FOR CUSTOMER
407 PLACE OUTBOUND CALL TO CUSTOMER
409 COMMUNICATE BILLING PRODUCTS TO CUSTOMER
A
410 RETRIEVE ADDITIONAL VALIDATION INFORMATION FROM CUSTOMER FOR INTERNAL DATABASE
A
411 IMPLEMENT SELECTED PRODUCT
412 SEND BILL TO CUSTOMER
413 BEGIN MONITORING BILL STATUS
414 HAS CUSTOMER PAID BILL WITHIN ALLOTTED TIME?
YES
NO
415 BLOCK/DENY CUSTOMER REQUESTS FOR TRANSACTION
SEND CUSTOMER PAST DUE/DELINQUENT BILL NOTICE
416
417 HAVE NUMBER OF NOTICES EXCEEDED PREDETERMINED MAXIMUM?
NO
YES
418 SUBMIT ACCOUNT TO BILL COLLECTION
419 CHANGE CUSTOMER ACCOUNT STATUS TO "PAID"
420 DOES CUSTOMER PAYMENT HISTORY MERIT CHANGE IN CUSTOMER SCORE?
NO
421 NO ACTION
YES
422 ADJUST CUSTOMER SCORE ACCORDING TO PAYMENT HISTORY

FIG. 5

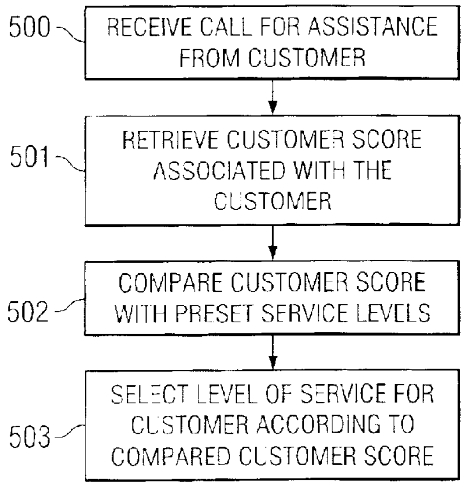
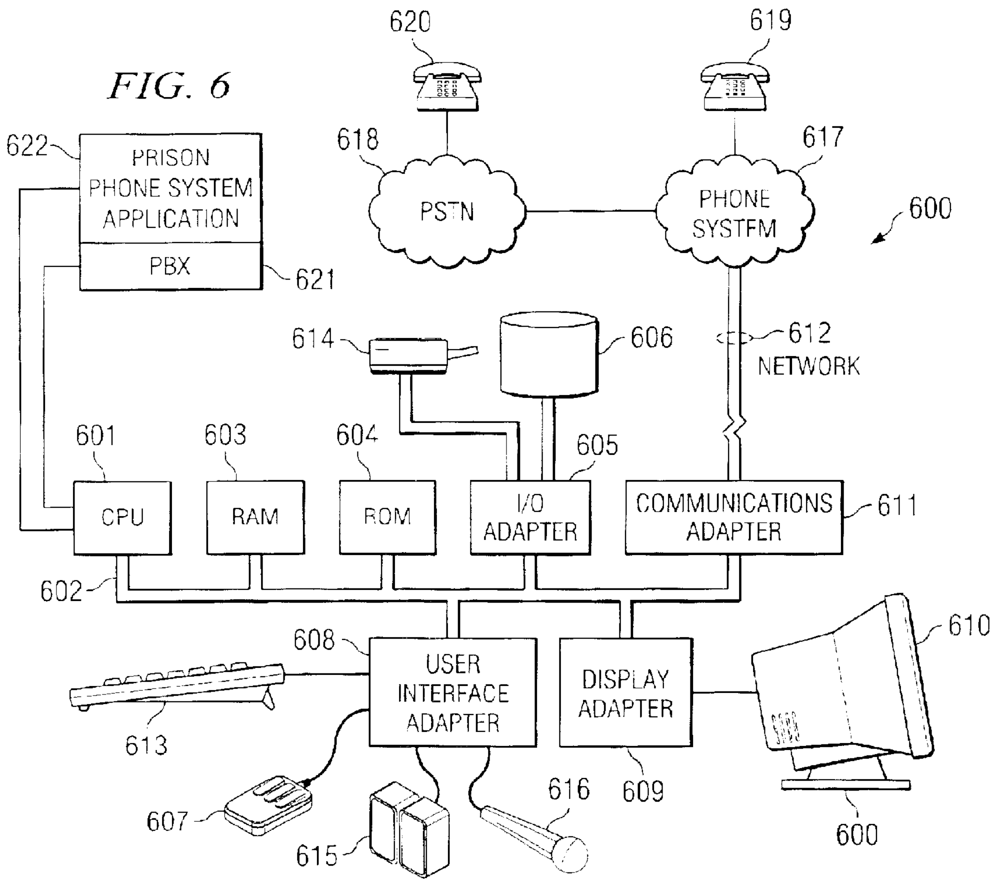

500
RECEIVE CALL FOR ASSISTANCE FROM CUSTOMER
501
RETRIEVE CUSTOMER SCORE ASSOCIATED WITH THE CUSTOMER
502
COMPARE CUSTOMER SCORE WITH PRESET SERVICE LEVELS
503
SELECT LEVEL OF SERVICE FOR CUSTOMER ACCORDING TO COMPARED CUSTOMER SCORE

FIG. 6


620
619
618
617
622
PRISON PHONE SYSTEM APPLICATION
PBX
621
PSTN
PHONE SYSTEM
600
614
606
612
NETWORK
601
603
604
605
CPU
RAM
ROM
I/O ADAPTER
COMMUNICATIONS ADAPTER
611
602
608
USER INTERFACE ADAPTER
DISPLAY ADAPTER
610
613
607
615
616
609
600

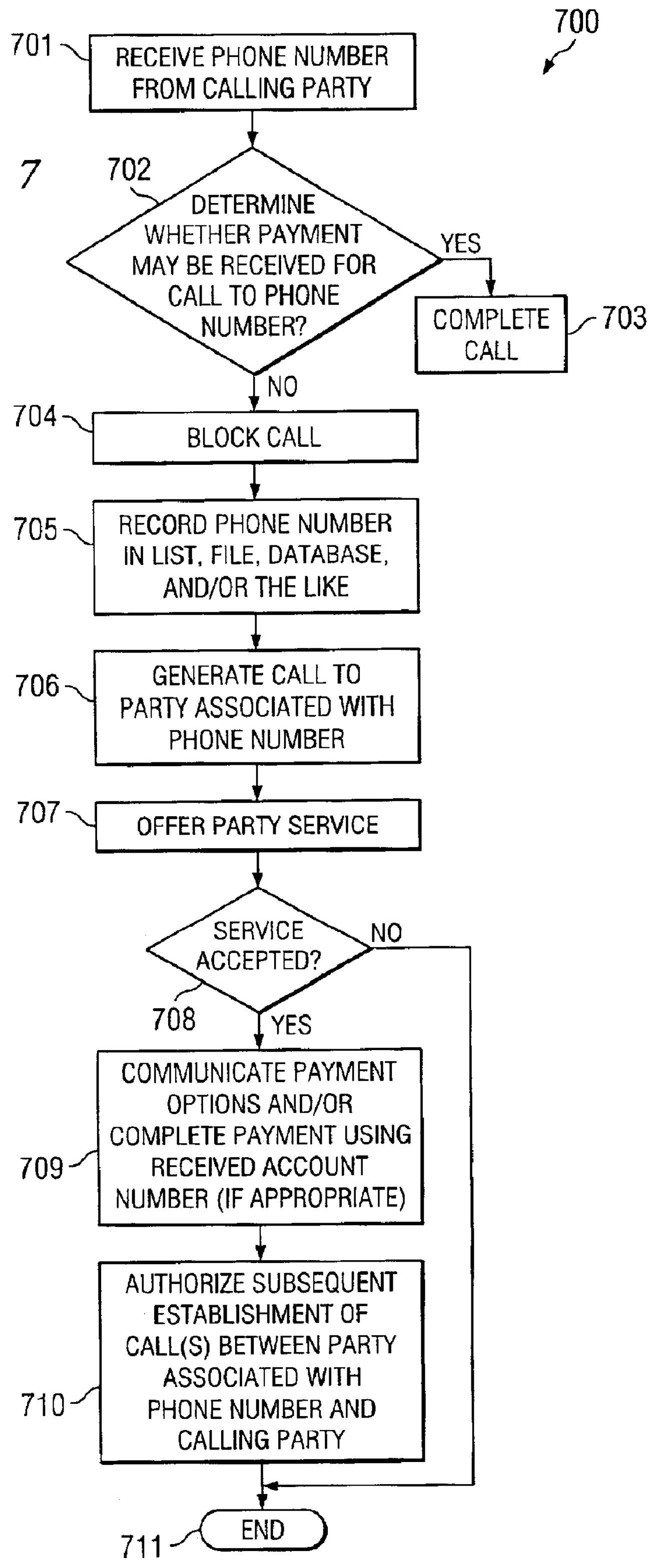
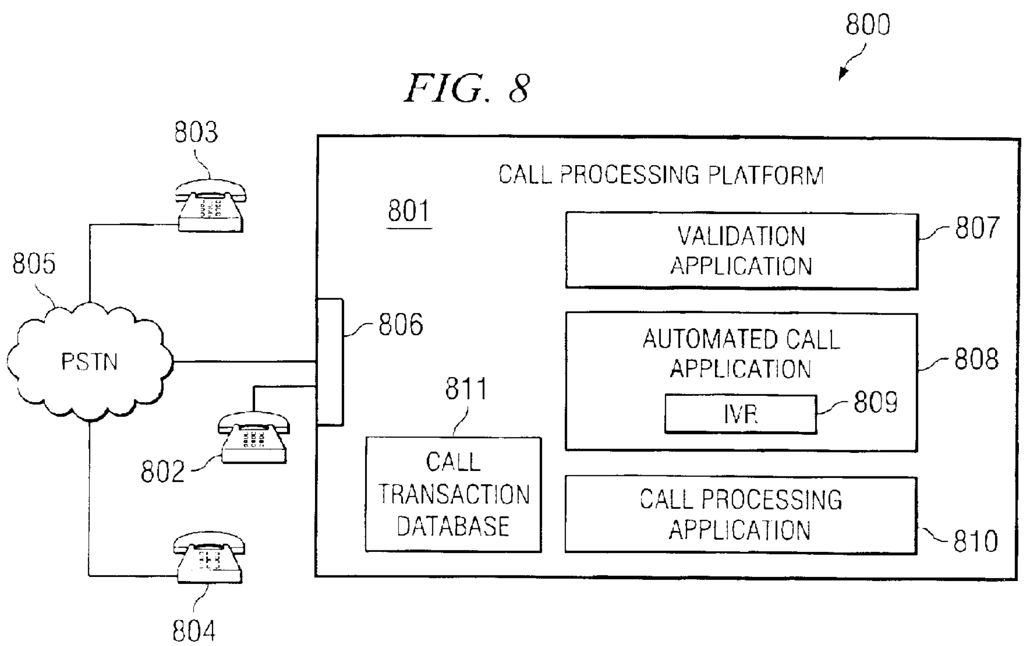

700
701
RECEIVE PHONE NUMBER FROM CALLING PARTY
702
DETERMINE WHETHER PAYMENT MAY BE RECEIVED FOR CALL TO PHONE NUMBER?
YES
COMPLETE CALL
703
NO
704
BLOCK CALL
705
RECORD PHONE NUMBER IN LIST, FILE, DATABASE, AND/OR THE LIKE
706
GENERATE CALL TO PARTY ASSOCIATED WITH PHONE NUMBER
707
OFFER PARTY SERVICE
SERVICE ACCEPTED?
NO
708
YES
709
COMMUNICATE PAYMENT OPTIONS AND/OR COMPLETE PAYMENT USING RECEIVED ACCOUNT NUMBER (IF APPROPRIATE)
710
AUTHORIZE SUBSEQUENT ESTABLISHMENT OF CALL(S) BETWEEN PARTY ASSOCIATED WITH PHONE NUMBER AND CALLING PARTY
711
END

FIG. 7

FIG. 8


800
CALL PROCESSING PLATFORM
801
VALIDATION APPLICATION
807
AUTOMATED CALL APPLICATION
808
IVR
809
CALL PROCESSING APPLICATION
810
CALL TRANSACTION DATABASE
811
806
PSTN
805
803
802
804

SYSTEMS AND METHODS FOR OFFERING A SERVICE TO A PARTY ASSOCIATED WITH A BLOCKED CALL

CROSS-REFERENCE TO RELATED APPLICATIONS

The present invention is continuation-in-part of U.S. patent application Ser. No. 10/135,883 filed Apr. 29, 2002 entitled "OPTIMIZING PROFITABILITY IN BUSINESS TRANSACTIONS"; the present application is also related to co-pending and commonly assigned United States patent application entitled "SYSTEM AND METHOD FOR REVERSE BILLING OF A TELEPHONE CALL," and to copending and commonly assigned U.S. patent application Ser. No. 10/135,878 entitled "INFORMATION MANAGEMENT SYSTEM AND METHOD," concurrently filed herewith, the disclosures of each of which are hereby incorporated by reference in their entirety.

TECHNICAL FIELD

The present invention relates generally to offering a service to a party associated with a dialed number, and more particularly, to a system and method for authorizing collect calls to a party associated with a dialed number.

BACKGROUND OF THE INVENTION

The generation of revenue and profitability is the driving force behind most business models. To supplement the cash purchasing methods in today's credit-based society, most businesses depend on some form of credit or entitlement authorization mechanism allowing for customers to purchase products, services, or other such items without the immediate physical exchange of cash. Inherent in such business models is the reality that a percentage of parties who purchase on credit or entitlement authorizations may eventually not pay, thus, diminishing the business' overall profitability.

In order to balance the risk of such losses against the benefits of maintaining credit entitlement systems, businesses go to great lengths to pre-screen credit applicants with lengthy applications requiring a wealth of personal information. This process is often-times slow and many consumers may decide to take their business to a competitor rather than wait for the completion of the credit application process. Such verification methods maximize risk prevention, but are incompatible with situations that require more immediate determinations.

One example of a business that requires more immediate credit/authorization determinations is the telecommunication provider industry, and, more particularly, businesses that provide telecommunication services to controlled-environment facilities, such as prisons. Prisoners are generally given some form of access to telephones, but the calls must be paid for. Prisons typically do not allow inmates to receive calls, thus, most incoming calls that are not directed to prison administration numbers are blocked. Moreover, prisoners, in general, do not have ready access to cash; therefore, calls are typically made collect.

As with other credit/authorization systems, some of the collect calls may never be paid for by the called parties. In such circumstances, the telecommunication service provider fails to recover the costs of providing the call, which, in turn, causes a loss of profitability. Bad debt losses may sometimes reach into the tens of millions of dollars for each telecommunication service provider with the industry total well over

$1 Billion. To address the risk of loss on some of the attempted correctional facility calls, telecommunication service providers sometimes obtain information on the called parties in order to establish a customer database for providing call verification/authorization. When an inmate attempts to make a collect call, the call or transaction request goes through a validation process. The telecommunication service provider accesses its customer databases and may be able to determine (1) can this call be billed (i.e., is there a billing arrangement with the local exchange carrier (LEC) or the called party), (2) if the destination number is already in the service provider's files, has the allotted credit limit been reached, and (3) has there been any information received from the LEC indicating that the called party has not been paying its bills. Depending on the extensiveness of the service provider's internal resources, the service provider may not be able to determine all three of these validation criteria. If favorable information is retrieved for each of the available validation criteria, the call is completed.

Conversely, if the inmate attempts to call a destination number that is not already on the customer database, or negative information is retrieved from the validation process, the service provider typically blocks the call from being completed. While these blocked calls save the telecommunication provider from losses for unpaid calls, some of those dropped calls represent lost potential revenue and profit that the provider would have generated.

Additional considerations that effect the revenue stream of telecommunication providers for prisons arise in the billing and collection (B&C) process. In providing collect calls, the service provider typically sends the collect call bill to the LEC that services the called number. LECs, such as Southwestern Bell, Verizon, BellSouth, Ameritech, and the like generally maintain accurate billing, name, and address (BNA) information, and may be authorized to bill third-party-provided telecommunication services if billing arrangements exists. It should be noted that for purposes of this disclosure, LEC is intended to include not only local exchange carriers, but also competitive LECs (CLECs), inter-exchange carriers (IXCs), and the like. LECs typically bill on a thirty-day billing cycle (i.e., provide a post-pay system that bills each customer for the telephone activity that occurred over the last thirty days). As with every other credit transaction, some LEC customers may fail to pay their bills. When this happens, the LECs recover any costs for providing the prisoner's call directly from the prison telecommunication service provider. Thus, the service provider carry all of the losses, which generally effects profit realization.

Moreover, because of the LECs' typical thirty-day billing cycle, the prison telecommunication provider may not become aware that the bill has become delinquent for a minimum of 120 days after the bill was originally sent to the LEC (LECs may not declare a particular bill uncollectable for 120 days or more in many circumstances). Thus, the service provider would not know to block further calls to that destination number for anywhere from four months to over a year. If calls continue to the delinquent destination number during that period, a substantial amount of revenue and profits would simply be lost.

A separate billing-related issue arises with LECs that do not have billing arrangements with the prison telecommunication service providers. If no billing arrangement exists with the LEC, the service provider must resort to billing the called party directly. In many circumstances, the service provider will not have accurate BNA information on the called party. The service provider may have to purchase this

information from the LEC. Additional costs may then be expended generating the direct bill. Therefore, the costs of this "random" direct billing may exceed the actual value recovered in some cases, which decreases profits even further. In response, many of the current systems simply choose to block all calls to destination numbers serviced by non-contracting LECs in order to alleviate this problem. Thus, as with the calls blocked due to a failure to achieve immediate positive validation, the calls blocked due to nonexistent billing arrangements with certain LECs may save some lost revenue and profits, but still represent potential lost profits at the point of demand for good paying customers.

Therefore, even though the business model of the prison telecommunications provider is centered on generating profits and recovering revenue, the service provider must account for the potential profit losses from (1) calls to destination numbers that are not blocked but which are ultimately not paid for; (2) continued calls to the same destination number that are allowed before the service provider becomes aware of delinquencies; (3) calls to destination numbers that may represent good credit risks and profit margins but which do not pass the initial validation process; (4) calls to destination numbers that represent good credit risks and profit margins that are nonetheless blocked because the destination number is serviced by LECs without billing agreements with the service provider. In addition to these different means for the service provider to lose revenue and eventually profits, service providers often contract with the prison or other such control-environment facility to pay the prison a commission on the value of each call provided for the privilege of providing the service to the prison. Because the commission typically is taken from the value of the call rather than the amount collected for the call, the commission will generally have to be paid regardless of whether the service provider actually ever collects the cost of the call. Many other different opportunities for revenue loss, such as governmental regulations, contractual commitments, and the like, also must be accounted for by these prison telecommunications providers.

BRIEF SUMMARY OF THE INVENTION

The present invention is directed to a system and method for optimizing profitability in business transactions. In one embodiment, the present invention is directed to a method of offering a service to a user. The method comprises receiving a dialed number from a request to initiate a collect telephone call from an origination source; processing the dialed number to determine whether payment may be received for the collect telephone call, wherein the processing blocks the collect call and stores the dialed number when the processing determines that payment may not be received for the collect telephone call; and establishing a telephone connection with the dialed number to communicate an offer for the service to a user associated with the dialed number. In another embodiment, the present invention is directed to a call processing platform. The call processing platform comprises an interface for receiving a dialed number from a request to initiate a collect telephone call from an origination source; a validation application for processing the dialed number to determine whether payment may be received for the collect telephone call, wherein the validation application is operable to block the collect call and to store the dialed number when the validation application determines that payment may not be received for the collect telephone call; and an automated call application that is operable to retrieve the stored dialed number to generate a telephone connection with the dialed number, wherein the automated call application communicates an offer for a service to a user associated with the dialed number.

The foregoing has outlined rather broadly the features and technical advantages of the present invention in order that the detailed description of the invention that follows may be better understood. Additional features and advantages of the invention will be described hereinafter which form the subject of the claims of the invention. It should be appreciated by those skilled in the art that the conception and specific embodiment disclosed may be readily utilized as a basis for modifying or designing other structures for carrying out the same purposes of the present invention. It should also be realized by those skilled in the art that such equivalent constructions do not depart from the spirit and scope of the invention as set forth in the appended claims. The novel features which are believed to be characteristic of the invention, both as to its organization and method of operation, together with further objects and advantages will be better understood from the following description when considered in connection with the accompanying figures. It is to be expressly understood, however, that each of the figures is provided for the purpose of illustration and description only and is not intended as a definition of the limits of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention, reference is now made to the following descriptions taken in conjunction with the accompanying drawing, in which:

FIG. **1** is a high-level block diagram illustrating the main elements of one embodiment of the present invention;

FIG. **2** is a high-level block diagram illustrating a more detailed view of the main elements of the embodiment shown in FIG. **1**;

FIG. **3** is a flowchart illustrating steps that may be used to implement the intelligent customer scoring system used in the embodiment of the present invention shown in FIGS. **1** and **2**;

FIG. **4** is a flow chart that represents the steps of functional interaction between the iROM and the iDBS, as shown in FIGS. **1** and **2**;

FIG. **5** is a flowchart illustrating steps that may be used to implement the intelligent high-value customer system used in the embodiment of the present invention shown in FIGS. **1** and **2**;

FIG. **6** depicts a block diagram of a computer system which is adapted to use the present invention;

FIG. **7** depicts an exemplary flowchart for offering a service to a user associated with a phone number of a previously blocked collect call according to embodiments of the present invention; and

FIG. **8** depicts an exemplary call processing platform according to embodiments of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Turning now to the drawing, FIG. **7** depicts flowchart **700** for offering a service to a user associated with a phone number of a previously blocked collect call according to embodiments of the present invention. In step **701**, a phone number of a party to be called may be received from a calling party or an origination source.

In step **702**, a logical comparison is made to determine whether payment may be received for the call to the received

phone number. The determination whether payment may be received may be based on any one or a number of factors. The determination may include determining whether the phone number is serviced by a telephone service provider (e.g., a local exchange carrier (LEC)) that permits billing to the party associated with the phone number by the party completing the collect call via the account associated with the phone number. The determination may involve determining the credit worthiness of the party associated with the phone number. If it is determined that the payment will most likely be received, the process flow proceeds to step **703** where the call is completed.

If it is determined that payment may not be received, the process flow proceeds to step **704** where the call is blocked. In step **705**, the received phone number is recorded for future processing. In embodiments of the present invention, the received phone number may be stored in association with creation of a call record. The received phone number may be stored in a suitable database, file, and/or the like. In step **706**, the database, file, or the like may be accessed to retrieve the phone number and a call is generated to a party associated with the phone number. The generation of the phone call may occur in a manual or autonomous manner. For example, an automated call distributor (ACD) may be utilized to generate a call to the phone number and to connect a customer service representative to the answering party. Alternatively, an automated call application may be utilized to generate a call the phone number and an interactive voice response (IVR) application may be utilized to communicate with the answering party.

In step **707**, the respective service (e.g., enabling the answering party to receive a subsequent collect call from the calling party) is offered to the user. In step **708**, a logical comparison is made to determine whether the service was accepted. If the service is not accepted, the process flow proceeds to step **711** where the process flow ends. If the service is accepted, the process flow proceeds to step **709**.

In step **709**, payment options may be communicated to the answering party. The service may include a direct billing arrangement to permit the answering party to receive a subsequent collect phone call or calls at the received phone number from the calling party. Alternatively, the service may include a prepayment plan to permit the answering party to receive the subsequent call or calls. In embodiments of the present invention, the user may communicate an account number to facilitate payment pursuant to a prepaid service. The payment for the prepaid service may be completed by initiating an electronic transfer of funds (e.g., from a bank account) or by debiting a credit account (e.g., a credit card account). In other embodiments, the completion of the payment may occur at a later time. The party may be allowed to make a payment through a third-party financial intermediary or the party may dial a suitable "1-800" number or the like to complete payment for the service.

In step **710**, subsequent collect calls from the calling party to the phone number are authorized. The subsequent collect call or calls may be permitted by establishing an appropriate telephony prepaid account. The account may be managed by a suitable call processing platform. In step **711**, the process flow ends.

FIG. **8** depicts exemplary system **800** that may be utilized to offer services according to embodiments of the present invention. System **800** comprises call processing platform **801**. Call processing platform **801** may be implemented, for example, as a service control point (SCP) to process collect calls from subscriber device **803** associated with public switched telephone network (PSTN) **805**. Call processing platform **801** may receive suitable messaging via SS7 signals to process the collect call. Alternatively, call processing platform **801** may receive a call connection from subscriber device **803** (e.g., at a 1-800 number) to process the collect call. In embodiments of the present invention, call processing platform **801** may be implemented as a telephony system at a controlled facility such as a correctional institution. Subscriber device **802** may be communicatively coupled to call processing platform **801** at the controlled facility.

Call processing application **810** may receive the phone number of the called party (in this example, subscriber device **804**) via interface **806** utilizing a suitable communication protocol. Call processing application **810** may communicate the phone number to validation application **807**. Validation application **807** determines whether payment may be received. Validation application **807** communicates the determination to call processing application **810**. Depending on the determination, call processing application **810** may connect subscriber device **803** or **804** to subscriber device **804** or block the call. Call processing application **810** may create a call record in call transaction database **811**. The call record may include identification of the origination source, the called phone number, whether the call was connected or blocked, the length of the phone call (if applicable), and/or the like.

Automated call application **808** may examine call records stored in call transaction database **811**. Automated call application **808** may determine that a collect call to subscriber device **804** was blocked. Automated call application **808** may generate a call to subscriber device **804**. If a party associated with subscriber device **804** answers, automated call application **808** may execute a call script utilizing IVR **808**. IVR **808** may be utilized to communicate the nature of the offered service (e.g., a prepaid or direct billing plan for subsequent collect calls). The user may accept or reject the communicated service. In embodiments of the present invention, the user may communicate an account number to facilitate payment for the prepaid option. IVR **808** may complete payment utilizing user input during the call connection or at a later time. IVR **808** may then communicate that the subsequent collect call(s) will be connected. IVR **808** may create a suitable prepaid account to permit the connection of the collect call(s). Call processing application **810** may then connect subsequent collect calls to subscriber device **804**.

Moreover, the present disclosure relates to systems and methods for optimizing profitability in business transactions including transactions related to call processing. It is understood, however, that the following disclosure provides many different embodiments, or examples, for implementing different features of the invention. Specific examples of components and arrangements are described below to simplify the present disclosure. These are, of course, merely examples and are not intended to be limiting. In addition, the present disclosure may repeat reference numerals and/or letters in the various examples. This repetition is for the purpose of simplicity and clarity and does not in itself dictate a relationship between the various embodiments and/or configurations discussed.

Turning now to FIG. **1**, intelligent call and billing management solution (iCBS) **10** comprises a multi-application system for optimizing and maximizing profitability of business transactions. ICBS **10** includes intelligent customer scoring system (iCSS) **100** for establishing a customer score, and intelligent revenue opportunity module (iROM) **101** for processing the customer score received from iCSS **100** to

produce a recommended revenue opportunity application. The purpose behind the calculation of a customer score may vary from application to application. In some embodiments of the present invention, a customer score may represent a predicted risk management score used to authorize or deny requested transactions. In other embodiments, a customer score may represent a profitability value of that customer. ICBS **10** also includes intelligent direct billing system (iDBS) **102**, associated with iROM **101**, for selecting any one of a number of direct billing products for a customer responsive to customer score-based determinations made within iROM **101**. In order to facilitate the calculations and predictive risk assessments made by iCSS **100**, iCBS **10** also includes billing and customer database (BCD) **104**. BCD **104** is generated as iCBS **10** continues interactions and completing transactions for customers of the business models. As more information is gathered with regard to the customer, it is stored in BCD **104** for future use and evaluation of the customer scores. BCD **104** is "local" to the prison telecommunication service provider, but may be physically located at the same facilities, different facilities, in a local area network (LAN), a wide area network (WAN), or other such network under the control or influence of the service provider.

It should be noted that in additional embodiments of the present invention, iCBS **10** may also incorporate intelligent high value customer service (iHVC) **103** for providing a variable and selective customer service options in which the selection of the level of service offered to a customer is made responsive to the customer score.

While the inventive elements of iCBS **10** are intended to be applicable to a wide variety of business models and business situations, the detailed description presented below of one embodiment of iCBS **10** will be specifically tailored to the example of a prison telecommunications service provider. Considering the specific example, as prisoners attempt to make out-going calls from the prison, the called phone number is received at iCBS **10** and processed by iCSS **100** for calculation of the customer score. The called number may be obtained in many different known ways, such as via dual tone, multiple frequency (DTMF) readers, via an Internet protocol (IP) network, via the signaling system **7** (SS7) network, or via dialed number identification service (DNIS) signals provided by the network.

If a particular called number is not already in the service provider's customer database, the customer score that is preferably calculated by iCSS **100** will be an initial score based on the immediately available local information associated with the requested call and the called number. The telecommunication service provider compares the initial customer score against a preset risk threshold to determine which calls will be allowed and which calls will be blocked. If a call is blocked, a message may preferably be played to the inmate informing him or her of the reason why the call was blocked, or switching the inmate to an operator or interactive voice response (IVR) unit to make arrangements for the requested or future calls. An outgoing call may also preferably be placed by the IVR system to the blocked called party. The outgoing call preferably informs the called party that an inmate was attempting to reach the called party but was blocked, and offers the called party alternative products for arranging future calls from the inmate. The called party may also preferably be asked for additional or supplemental detailed information that may be used for refining the customer score. For example, the called party may be asked for their social security number, the length of time at their home/job, whether they rent or own, and the like. All of this information is preferably stored in BCD **104** for future use. For security purposes, the outgoing call may also preferably offer the called party an option to permanently block the called party's number from being called. Such circumstances may be necessary to thwart harassment of victims by the inmates.

In the prior art systems, the calls that were blocked are completely lost. However, in the described embodiment of the present invention, iROM **101** may preferably determine the profit opportunities that may exist in the blocked calls. ICBS **10** preferably communicates the called numbers of blocked calls **105** to iROM **101** for rating the related customer scores. An opportunity risk matrix (ORM) is established by the service provider identifying various service or billing plans aimed at optimizing profit opportunities for some of the blocked calls. The different risk levels found within the ORM may be lower than the threshold risk level associated with the real-time call authorization procedure. However, the varying levels are selected to correspond to the various customer scores (e.g., a higher risk customer score may be paired with a low-risk billing and collection method, while a low risk customer score may be associated with a higher risk billing and collection plan). As iROM **101** processes the customer score in comparison to the ORM levels through a software or hardware product or application comparator, a subset of possible revenue opportunity applications is identified as potentially available to the customers according to their respective customer scores. These possible revenue opportunity applications may either be presented to the customer for selection during one of the outbound calls made from the IVR system, or may be selected and implemented by the service provider directly.

In addition to blocked calls **105**, that are blocked based on an initial validation failure, other blocked calls may exist which are blocked for different reasons. For example, in billing situations, if the prison telecommunications service provider does not have a billing arrangement with a LEC, then the service provider generally cannot bill the LEC for any of the calls that were made to destination numbers serviced by that particular LEC. As noted previously, many of the prior art systems simply block all calls that were made to destination numbers serviced by those non-contracting LECs. In contrast, the described embodiment of the present invention will preferably communicate LEC unbillable calls **106** to iROM **10** for determining whether a direct billing method would be desirable and/or possible for increasing the potential profits from those previously blocked or lost LEC unbillable calls **106**. Other blocked calls are also communicated to iROM **101** for determining whether a revenue opportunity application or product would be desirable for potentially increasing profits and revenue recovery on similar blocked calls.

In making this determination of possible revenue opportunity applications and products, parameters such as customer score or risk metric, the general length of the inmate's incarceration (which may preferably be obtained through deduction based on the prison unit from which the call originates, which, itself, may be deduced from the area code and prefix of the calling number, typically obtained from the automatic number identification (ANI) service), the potential profit margin available for the type of calls predicted (e.g., long distance vs. local or comparing average call costs vs. recovery margins for different geographic regions or comparing the available billable rates), the payment history of a particular customer or owner of the destination number, contractual obligations, commission agreements, governmental regulations, the type of transaction requested, costs

for external validation, billing and collection (B&C) costs, rate revenue per call, the cost of the different billing products versus their economic benefits, and the like are considered. By weighing those and other parameters in the iT brain or system processor, iROM **101** preferably selects a variety of different payment products, which may include products such as standard post-paid products or prepaid products. They may also preferably determine a billing cycle, such as every 30 days, every two weeks, or even every few days depending on the customer score. They may also preferably determine the appropriate payment cycle, such as 30 days, two weeks, or a couple of days from receiving the bill. IROM **101** also preferably determines whether a direct billing product is necessary and/or desirable, or whether LEC billing product is acceptable. For LEC unbillable calls **106**, a direct billing system would be desirable as the prison telecommunication service provider is not capable of billing that particular LEC without a billing arrangement.

For example, when calls are blocked, iCBS **10** may preferably make an automated call to the intended destination number to inform the called party that the inmate's attempted collect call was blocked. The system may then provides the called party several options that may allow the inmate's future calls to that number to be completed. However, instead of calling all of the blocked calls, the system preferably evaluates the customer score of the destination number, as well as the additional information described above, to selectively determine which of the blocked called parties represent attractive profit opportunities. Therefore, instead of losing all of the revenue that could be made from the blocked calls, the described embodiment of the present invention preferably optimizes the potential recovery of revenue and ultimate profitability.

As part of the general management functions performed by the inventive system, the transaction authorizations may take the form of a credit-type authorization, in which the customer would be limited to a certain monetary value of credit, or an entitlement-type authorization, in which the customer would be limited to a numerical, per-use entitlement. In administering either type of usage authorization system, the present invention intelligently sets the usage limit based on individual customer segmentation using the customer score and the additional information considered by iCBS **10** to manage the risk of the transaction.

In the current embodiment described in the context of a prison telecommunication service provider, the authorization would typically be a predictive risk management system. Therefore, depending on the customer score of a particular called party, the inmate would be capable of making calls to that called party up to an individually tailored limit on a transaction-by-transaction basis. For example, considering an initial customer score, because it represents a first transaction request for that called number, the system may set a lower usage limit for the allowed costs of the single call. If, during the course of the call, the usage limit is reached, the called party may either be disconnected, or provided a message and options for paying to extend the call. Similarly, considering a refined customer score, depending on the actual score, a called party could have a usage limit of anywhere from $20.00 to $800, depending on the level of investment the service provider is willing to accept or make in the customer. Furthermore, as a customer continues to build a balance closer to the determined usage limit, iCSS **100** will increase the risk level reflected in the customer or profitability score to reflect the increased investment.

Turning now to FIG. **2**, FIG. **2** is a high-level block diagram depicting a more detailed view of some of the elements of iCBS **10**, as shown in FIG. **1**. In an example operation of the described embodiment of the present invention as applied to a prison telecommunications provider, the elements of FIG. **2** illustrate the system configured to implement and execute the described embodiment. In operation, an inmate enters phone number/transaction request **20** into a prison phone, which requests the transaction of a collect call. Phone number/transaction request **20** is received by iCSS **100** and processed at intelligent targeting (iT) brain **200** to preferably determine a predictive risk level associated with phone number/transaction request **20**.

Assuming, for purposes of this example, that phone number/transaction request **20** is the first transaction request for this particular called number, iT brain **200** preferably makes a real-time determination to authorize or block the requested call. To make the real-time authorization, iT brain **200** preferably accesses internal database **201** to find generic demographic information related to phone number/transaction request **20**. By using generic and widely available demographic information, as correlated to phone number/transaction request **20**, iT brain **200** preferably calculates the initial customer score for purposes of authorizing or blocking the initial collect call request. In additional embodiments, iT brain **200** may also initially access the local information which may provide accurate BNA information, the general length of the inmate's incarceration, the potential profit margin available for the type of calls requested, the payment history of a particular customer or owner of the destination number, contractual obligations, commission agreements, governmental regulations, the type of transaction requested, costs for external validation, B&C costs, rate revenue per call, rate of consumption, and the like. IT brain **200** may preferably cross-reference census data that has been associated with area codes and telephone number prefixes, by using the Bellcore/Telecordia area code system, the NPA/Nxx numbering system. The related census data may preferably be stored locally and used by iT brain **200** to find generalized income levels of persons residing in that geographic area, average home values of persons residing in that area, and, therefore, facilitate calculating an intelligent real-time customer score based solely on the use of the NPA/Nxx data.

After making its real-time determination based on the information gathered from internal database **201**, iT brain **200** preferably assigns customer score **202** to phone number/transaction request **20**. Customer score **202** will typically reside within preset risk range **204**. Depending on the system used by prison telecommunication provider, a high score may correlate to a low risk, high profitability transaction request, whereas a low score would represent a high risk, low profitability transaction request. It should be noted that other scoring systems with different representations of risk levels may also be used in preset risk range **204**. Customer score **202** is then compared by iT brain **200** with risk threshold **203**. Risk threshold **203** may also preferably be pre-selected by the prison telecommunication service provider on a sliding scale represented by threshold range **205**. If the risk level associated with customer score **202** represents a better risk than risk threshold **203**, iCSS **100** allows phone number/transaction requests **20** to be completed. However, if customer score **202** represents a higher risk value than risk threshold **203**, phone number/transaction request **20** is denied or blocked.

After iT brain **200** makes the determination of whether to block or connect phone number/transaction request **20**, it begins to refine customer score **202** by accessing external databases **21-1** through **21**-N. The process of accessing

external databases **21-1** through **21**-N is sometimes referred to as data mining. IT brain **200** mines for data associated with phone number/transaction request **20** that would indicate a tendency for a low or high customer score. Such information may include payment histories for certain on-line vendors, in-service dates for the destination phone number, the type of dwelling that is serviced by the destination telephone number (i.e., whether the dwelling is a home or an apartment), accurate BNA information, and the like. IT brain **200** will preferably mine data in external databases **21-1** through **21**-N and possibly find correlation elements **22** to correlate the data mined with the destination of phone number preferably within a period of time after the initial collect call being made.

After completion of the data mining process and evaluating all of correlation elements **22**, iT brain **200** preferably recalculates a predictive risk management value and refines customer score **202** to phone number/transaction request **20**. As opposed to the real-time determination represented by initial customer score **202**, refined customer score **202** preferably represents a more accurate investigation and prediction into the customer score or predictive risk of the called party at phone number/transaction request **20**. The eventual level of customer score **202** may preferably indicate to iCSS **100** that future calls should either be blocked or be turned over to iROM **101** for determination of a revenue opportunity product to be offered to the called party

When a revenue opportunity product may be desirable, iCSS **100** preferably communicates customer score **202** to iROM **101** for determining appropriate revenue or profit opportunity products or applications to offer or select for the called party. IROM **101** examines customer score **202** and compares it to the ORM to select various payment and billing products according to cycle list **207** and product list **208**. The service provider would either present the list of possible products to the customer for selection, or select the most desirable product automatically. For example, if phone number/transaction request **20** is serviced by a LEC which does not have a billing arrangement with the prison telecommunication service provider, yet its customer score **202** represents an extremely low risk and high profitability margin, iROM **101** may select to offer standard payment products, which include post-paid products billed directly by the service provider on a monthly payment cycle. These high-valued customers may also preferably warrant additional grace periods in which to make payments after the 30-day cycle. In contrast, if customer score **202** represents a higher risk value than risk threshold **203**, but still offers a reasonable predictive risk of recovering revenue, iROM **101** may select to offer different prepaid calling packages to the called party at phone number/transaction request **20**, including various denominations and accounts to pay for future or pending collect calls. Additional, if customer score **202** represents a high predictive risk, but still an acceptable level, instead of a prepaid product, a standard post-paid bill product may be offered. However, instead of providing for a 30 day billing cycle as with the lower risk customers, iROM **101** may select more frequent billing cycles, such as billing every three days, with a payment return time of three days. In this manner, the higher risk individuals will be billed more quickly and, if their payments are not received in the time allotted, their accounts may be blocked before further costs are expended completing calls that may not be paid for.

In conjunction with the revenue opportunity products offered by iROM **101**, iDBS **102** adds direct billing methods to supplement the revenue opportunity products. For example, if phone number/transaction request **20** has a very low risk level in customer score **202**, but, the destination number is served by a LEC which does not have a billing agreement with the prison telecommunication provider, iCBS, as depicted in FIG. **1**, would still operate to optimize profits from such potential callers by implementing a direct billing scheme from iDBS **102**, which sends bills directly to the called party to pay for collect calls made from the prison. Thus, while the prior art methods would simply block all such calls, the method described in this embodiment of the present invention, operate to optimize profits that would otherwise go unrealized. It should be noted that additional necessity for direct billing may generally arise when iROM **101** designates or selects billing and/or payment cycles of less than the standard 30 day periods. Because typical LECs do not bill in cycles less than 30 days, it would be required that the prison telecommunications provider bill those clients directly for the shortened billing and/or payment cycle. Alternatively, for low risk, high profit margin customers, standard billing through LEC billing **23** could be continued. The ultimate choice of billing products preferably results from a comparison of the cost and economic benefits for the different available products.

Additional embodiments of the present invention may also include iHVC **103** to provide customer services. Even though the specifically described example addresses issues involved in serving prisons, a considerable amount of revenue is still available in providing such telecommunication services to the prison and the prisoners' family communities. Thus, as in any other service oriented business, customer service and customer satisfaction is important in maintaining revenue and also increasing the recovery of revenue and ultimate firm profitability. With these goals in mind, iCBS **10**, as depicted in FIG. **1**, provides a system for offering variable levels of service to customers depending on the value of customer score **202**. For example, average customers who pay a small amount on low margin accounts may be placed into a queue for an interactive voice response (IVR) system in order to access help applications or caller customer services. In contrast, highly valued customers with customer score **202** that represent very low predictive risk and very high profit margin, may be treated differently in the service center interactions. These higher valued customers may, for instance, be placed at the beginning of the queue to the IVR system. Alternatively, the higher valued customers may have a separate customer service access number to call that offers fewer and shorter waiting periods and possibly even more customer service opportunities. Even further, the highest valued customers may be directly connected to live operators in order to access the customer service system. In creating the different service levels in service level list **206**, iCBS **10** (FIG. **1**) operates to preserve and extend the possible revenue recovery from the highest valued customers, while at the same time providing incentive for the lower valued (i.e., higher predictive risk and lower profit margin) customers to increase their customer or profitability score to obtain higher levels of service.

FIG. **3** is a flow chart that illustrates the steps implemented in carrying out the function of the iCSS of the revenue optimizing system of the described embodiment of the present invention shown in FIGS. **1** and **2**. In step **300**, the iCSS receives a phone number and transaction request. The transaction request and/or related phone number may be an initial transaction request or may be a repeat transaction request. In step **301**, the system checks the local database for validation (validation may comprise the verification that a billing mechanism is in place to bill the requested

transaction). If the phone number cannot be validated, the information is forwarded to the iROM in step **302**. In step **303**, the local database is again checked for the telephone number. If the record of the phone number is not found in the local database, the system preferably checks for an initial customer score related to the telephone number in step **304**. In step **305**, the profitability of the transaction is determined. In step **306**, using both the profitability information and the initial customer score, the system determines whether the requested transaction is expected to be profitable. If so, the transaction is authorized in step **309**. If not, the transaction is denied in step **307** and then transferred to the iROM in step **308**.

While the initial customer score is a value calculated in real-time to make a real-time determination to authorize or deny a requested transaction, the refined customer score is a more detailed prediction of risk or dependability level as performed over a longer amount of time and is also a more detailed review of information associated with the phone number. When an initial transaction request has been made, as the iCSS makes its real-time customer assessment, it also begins the process of predicting and determining the refined customer score for subsequent transaction requests. If the phone number is found in the local database in step **303**, the database is checked for the existing refined customer score in step **310**. The profitability of the transaction is determined in step **305**, whereafter a determination is made in step **306** whether the requested transaction is expected to be profitable, based at least in part on the profitability information and the refined customer score. If it is expected to be profitable, the transaction is authorized in step **309**. If not, the transaction is denied in step **307** and the information is transferred to the iROM in step **308**.

If an initial customer score was calculated as a result of step **304**, the system preferably accesses external databases in step **313**, checks any customer payment experience in step **314**, and gathers additional customer-specific information in step **312**, after interacting with the customer in step **311**. For example, during any outbound calls placed by the system's IVR units to the called party, the IVR applications prompt the called party for additional, useful identification and validation information. Alternatively, the information may be retrieved when the called party calls the service provide either with live, voice-to-voice or chat capabilities. Using this additional information, the initial customer score is preferably refined into the refined customer score in step **315**. Refinement is a continuous process as new information is obtained. Thereafter, the refined customer score is stored in the local database for future use in step **316**. As shown in FIG. **3**, the iT brain essentially performs the functions described in steps **304–306**, and **310**.

FIG. **4** is a flow chart that represents the steps of functional interaction between the iROM and the iDBS, as shown in FIGS. **1** and **2**. The iROM is preferably accessed in response to the delivery of a particular customer score corresponding to a called telephone number or one or more different blocked calls in order to determine whether a revenue opportunity exists in providing one or more of the revenue opportunity products. In step **400**, the iROM receives a blocked or unpaid account along with the telephone number associated with that account. Upon receipt of that blocked account, the iROM retrieves the customer score (either initial or refined) associated with that account in step **401**. The iROM is generally implemented when blocked accounts are received; whether those blocked accounts are due to a bad debt or due to the destination numbers being serviced by a LEC without a billing arrangement with the prison communication provider. In step **402**, the iROM preferably compares the customer number and the transaction profitability with the ORM to identify possible applicable revenue or profit generation applications and billing and payment products.

The iROM determines whether a direct billing product would be available or beneficial to the customer in step **403** based on its cost and economic benefits. If no direct billing product is suggested, then, in step **404**, the iROM will pass the billing structure over to a standard third-party-billing-authority option, such as the LEC billing in the prison service provider example. If, on the other hand, it is determined that an appropriate direct billing product may be available from the iDBS, the iROM will identify and/or select a particular product or a set of possible products for a customer in step **405**. In step **406**, a determination is made whether the selected product or products are pre-paid products or post-paid products. If one or more of the selected products is a pre-paid product, the product is grouped for presentation to the customer in steps **407** and **408**. If, however, the selected product includes a post-paid product, the iDBS, responsive to the customer score comparison, will select the appropriate variable billing cycle and/or variable payment cycle for the customer in step **408**. The identified post-paid products may also be assembled with any of the identified prepaid products for presentation to the customer in steps **407** and **408**. However, the in the described embodiment, the system will preferably automatically select the specific direct billing product that is most beneficial for profit optimization. In step **407**, the system places an outbound call to the customer at the telephone number. During the call, the different identified prepaid and/or post-paid products are presented to the customer to select for facilitating future transactions in step **409**. Additionally, in step **410**, the system retrieves additional validation and/or identification data from the customer to supplement the information contained in the internal/local database. Upon receiving the product selection from the customer, the system will implement the selected product in step **411**.

If a post-paid product is selected, the system sends the bill directly to the customer in step **412**. Once the bill has been sent to the customer, the system preferably begins to monitor the bill status of the outstanding pending bill in step **413**. As a part of the monitoring process, in step **414**, the system determines whether the customer has fully paid the bill within the time allotted. If the customer has not fully paid the bill within the allotted time, the system preferably blocks or denies any future or pending requests of the customer in step **415**. The iDBS preferably sends the customer a past due or delinquent notice in step **416**. Depending on the number of past due or delinquent notices, as determined by step **417**, the customer's account may either be submitted to a bill collection service in step **418**, or passed back to the decision block of step **414** to determine whether the customer has paid the bill. As soon as the customer is detected to have paid the bill, the status on the customer's account is preferably changed to a "paid" status in step **419**.

To maintain an accurate predictive risk management level or customer score, the iT brain may dynamically adjust the profitability or customer score associated with the customer. In step **420**, in addition to gathering the additional information from both the internal and external databases, it is determined whether the specific payment history of the customer merits any change in the customer's related customer score. If no change is indicated, no action is taken in step **421**. However, if an adjustment is merited according to the payment history, the system preferably recalculates or

adjusts (i.e., refines) the customer score according to the customer's specific payment history in step **422**. For example, if the customer has numerous delinquent notices, and/or numerous late payment notices, the system may preferably adjust the customer score to reflect a higher predictive risk associated with that customer. Conversely, if the customer has exhibited a good bill payment history over a period of time, the system may preferably adjust the customer score to reflect a lower predictive risk value for that customer. Thus, as the iT brain learns more about the payment characteristics of the customer, the customer's customer score or profitability level will preferably be adjusted accordingly to accurately reflect the level of profitability or predictive risk for the customer.

FIG. **5** is a flow chart that represents steps that may be implemented when executing the functions of the iHVC of the described embodiment of the present invention shown in FIGS. **1** and **2**. In step **500**, the iHVC will receive a call for assistance from a customer. The call may be placed to a customer call center, or some other IVR system that implements a customer service call center. Once the call has been received, the iHVC retrieves the customer score associated with the particular customer in step **501**. In step **502**, the customer score will be compared against the matrix of preset service levels. The iHVC will then select a particular level of service for the customer responsive to the compared customer score, in step **503**. The resulting service provided to a customer will, therefore, vary according the customer's customer score.

It should be noted that the present invention is not limited to the described embodiment tailored for prison telecommunication service providers. Alternative embodiments of the present invention may be applied to other businesses as well. For example, remaining in the telecommunications industry, the collect call (i.e., 0+, 1-800-COLLECT, 1-800-CALL-ATT, etc.), 10-10-XXX long distance services, and the like may also implement the present invention to improve its capability for optimizing profitability. Outside of the telecommunication industry, credit and billing systems may be based on the present invention keying credit authorization and risk-dependent transactions on telephone numbers with variable billing and service procedures selectively tailored for the individual customer segment. Any system in which a telephone number may be used to designate the ultimately responsible party may implement various embodiments of the present invention.

Additionally, outside of the credit authorization sectors, call centers may be able to implement various embodiments of the present invention for selectively offering customer service to various callers based on the customer score for potential profitability. For instance, if a caller calls into a business call center, the customer score analysis may be performed to determine that the caller is likely a profitable potential customer. As a result of this customer score, the call center may direct the caller to a more personal calling experience to enhance the callers interaction with the call center.

When implemented in software, the elements of the present invention are essentially the code segments to perform the necessary tasks. The program or code segments can be stored in a processor readable medium or transmitted by a computer data signal embodied in a carrier wave, or a signal modulated by a carrier, over a transmission medium. The "processor readable medium" may include any medium that can store or transfer information. Examples of the processor readable medium include an electronic circuit, a semiconductor memory device, a ROM, a flash memory, an erasable ROM (EROM), a floppy diskette, a compact disk CD-ROM, an optical disk, a hard disk, a fiber optic medium, a radio frequency (RF) link, etc. The computer data signal may include any signal that can propagate over a transmission medium such as electronic network channels, optical fibers, air, electromagnetic, RF links, etc. The code segments may be downloaded via computer networks such as the Internet, Intranet, etc.

FIG. **6** illustrates computer system **600** adapted to use the present invention. Central processing unit (CPU) **601** is coupled to system bus **602**. The CPU **601** may be any general purpose CPU. However, the present invention is not restricted by the architecture of CPU **601** as long as CPU **601** supports the inventive operations as described herein. Bus **602** is coupled to random access memory (RAM) **603**, which may be SRAM, DRAM, or SDRAM. ROM **604** is also coupled to bus **602**, which may be PROM, EPROM, or EEPROM. RAM **603** and ROM **604** hold user and system data and programs as is well known in the art.

Bus **602** is also coupled to input/output (I/O) controller card **605**, communications adapter card **611**, user interface card **608**, and display card **609**. The I/O adapter card **605** connects to storage devices **606**, such as one or more of a hard drive, a CD drive, a floppy disk drive, a tape drive, to the computer system. The I/O adapter **605** is also connected to printer **614**, which would allow the system to print paper copies of information such as document, photographs, articles, etc. Note that the printer may a printer (e.g. dot matrix, laser, etc.), a fax machine, or a copier machine. Communications card **611** is adapted to couple the computer system **600** to a network **612**, which may be one or more of a telephone network, a local (LAN) and/or a wide-area (WAN) network, an Ethernet network, and/or the Internet network. User interface card **608** couples user input devices, such as keyboard **613**, pointing device **607**, and microphone **616**, to the computer system **600**. User interface card **608** also provides sound output to a user via speaker(s) **615**. The display card **609** is driven by CPU **601** to control the display on display device **610**.

When operating any one or number of embodiments of the present invention as shown in FIGS. **1** & **2**, and, in particular, the prison telecommunication service provider embodiment described herein, computer system **600** preferably connects via network **612** to phone system **617**, which is connected at some interface point to public switched telephone network (PSTN) **618**. Computer system **600** preferably includes software applications, private branch exchange (PBX) **621** and prison phone system application **622**, run from CPU **601** to control all aspects of the present invention. As inmates attempt to call from phone **619** to destination number/phone **620**, the embodiment of the present invention running on computer system **600** facilitates the connection and the probabilities that the service provider will recover the cost and revenue related to that call.

Although the present invention and its advantages have been described in detail, it should be understood that various changes, substitutions and alterations can be made herein without departing from the spirit and scope of the invention as defined by the appended claims. Moreover, the scope of the present application is not intended to be limited to the particular embodiments of the process, machine, manufacture, composition of matter, means, methods and steps described in the specification. As one of ordinary skill in the art will readily appreciate from the disclosure of the present invention, processes, machines, manufacture, compositions of matter, means, methods, or steps, presently

existing or later to be developed that perform substantially the same function or achieve substantially the same result as the corresponding embodiments described herein may be utilized according to the present invention. Accordingly, the appended claims are intended to include within their scope such processes, machines, manufacture, compositions of matter, means, methods, or steps.

What is claimed is:

1. A call processing platform, comprising:

an interface for receiving a dialed number from a request to initiate a collect telephone call from an origination source;

a validation application for processing said dialed number to determine whether payment may be received for said collect telephone call, wherein said validation application is operable to block said collect call and to store said dialed number when said validation application determines that payment may not be received for said collect telephone call; and

an automated call application that is operable to retrieve said stored dialed number to generate a telephone connection with said dialed number, wherein said automated call application communicates an offer for a service to a user associated with said dialed number.

2. The call processing platform of claim **1** wherein said automated call application comprises an interactive voice response system.

3. The call processing platform of claim **1** wherein said call processing platform communicates with an external database to receive credit information of said user associated with said dialed number.

4. The call processing platform of claim **1** wherein said validation application determines that said dialed number is serviced by a telephony service provider that does not permit billing to said user by a service provider that operates the call processing platform.

5. The call processing platform of claim **1** wherein said automated call application offers said user a billing arrangement to permit said user to receive a subsequent collect call from said origination source.

6. The call processing platform of claim **1** wherein said automated call application offers said user a prepaid service to permit said user to receive a subsequent collect call from said origination source.

7. The call processing platform of claim **6** wherein said automated call application prompts said user for account information to initiate payment for said prepaid service.

8. The call processing platform of claim **7** wherein said payment for said prepaid service is completed by said automated call application.

9. The call processing platform of claim **8** wherein said payment is made by an electronic transfer of funds from an account.

10. The call processing platform of claim **8** wherein said payment is made by debiting a credit account of said user.

11. A method of offering a service, comprising:

receiving a dialed number from a request to initiate a collect telephone call from an origination source;

processing said dialed number to determine whether payment may be received for said collect telephone call, wherein said processing blocks said collect call and stores said dialed number when said processing determines that payment may not be received for said collect telephone call; and

establishing a telephone connection with said dialed number to communicate an offer for said service to a user associated with said dialed number.

12. The method of claim **11** wherein said processing determines that said dialed number is serviced by a telephony service provider that does not permit billing to said user by a service provider that offers said service.

13. The method of claim **11** further comprising:

offering said user a billing arrangement to permit said user to receive a subsequent collect call from said origination source.

14. The method of claim **11** wherein said processing utilizes credit information related to said user associated with said dialed number that is received from an external database.

15. The method of claim **11** wherein said establishing a telephone connection is performed by an interactive voice response (IVR) system.

16. The method of claim **15** further comprising:

establishing a prepaid account to permit said user to receive a subsequent collect call from said origination source utilizing said IVR system.

17. The method of claim **16** further comprising:

prompting said user for account information to initiate payment for said prepaid service utilizing said IVR system.

18. The method of claim **17** further comprising:

completing said payment for said prepaid service.

19. The method of claim **18** wherein said payment is made by an electronic transfer of funds from an account.

20. The method of claim **18** wherein said payment is made by debiting a credit account of said user.

21. A call processing system, comprising:

means for receiving a dialed number from a request to initiate a collect telephone call from an origination source;

means for processing said dialed number to determine whether payment may be received for said collect telephone call, wherein said means for processing is operable to block said collect call and to store said dialed number when said means for processing determines that payment may not be received for said collect telephone call; and

means for generating a telephone connection with said dialed number, wherein said means for generating offers a service to a user associated with said dialed number.

22. The system of claim **21** wherein said means for generating comprises an interactive voice response (IVR) system.

23. The system of claim **21** wherein means for processing processes credit information of said user associated with said dialed number, wherein said credit information is received from an external database.

24. The system of claim **21** wherein said means for processing determines that said dialed number is serviced by a telephony service provider that does not permit billing to said user by a service provider that operates the call processing platform.

25. The system of claim **21** wherein said means for generating offers said user a billing arrangement to permit said user to receive a subsequent collect call from said origination source.

26. The system of claim **21** wherein said means for generating offers said user a prepaid service to permit said user to receive a subsequent collect call from said origination source.

27. The system of claim **26** wherein said means for generating prompts said user for account information to initiate payment for said prepaid service.

28. The system of claim **27** wherein said means for generating completes payment for said prepaid service.

29. The system of claim **28** wherein said payment is made by an electronic transfer of funds from an account.

30. The system of claim **28** wherein said payment is made by debiting a credit account of said user.

31. A call processing platform, comprising:

a validation application processing information regarding an attempt at establishing a desired telephone call and determining if completion of said desired telephone call meets operational objectives, wherein said validation application is operable to block said call when said validation application determines that completion of said desired telephone call does not meet said operational objectives; and

an automated call application generating an outbound call independent of said desired telephone call to communicate an offer for facilitating future telephone calls corresponding to said desired telephone call if said desired telephone call is blocked.

32. The call processing platform of claim **31** , wherein said operational objectives comprise:

an acceptable risk regarding whether payment may be received for said call.

33. The call processing platform of claim **31,** wherein said desired telephone call comprises a call to a particular dialed number and said outbound call communicating said offer is placed to said particular dialed number.

34. The call processing platform of claim **31,** wherein said communicated offer comprises at least one service plan selected from a plurality of service plans available for facilitating future telephone calls.

35. The call processing platform of claim **34,** wherein said at least one service plan is selected from said plurality of service plans based upon a risk level identified with said desired call.

36. The call processing platform of claim **31,** wherein said communicated offer comprises at least one billing plan selected from a plurality of billing plans available for facilitating future telephone calls.

37. The call processing platform of claim **36,** wherein said at least one billing plan is selected from said plurality of billing plans based upon a risk level identified with said desired call.

38. The call processing platform of claim **31,** wherein said desired call comprises a collect call and said outbound call comprises a call for which a recipient of the call does not incur charges.

39. The call processing platform of claim **31,** wherein said desired telephone call terminates in a controlled-environment facility.

40. The call processing platform of claim **39,** wherein said controlled-environment facility comprises a prison facility.

41. The call processing platform of claim **39,** wherein at least a portion of said call processing platform is disposed within said controlled-environment facility.

42. A method for processing calls, said method comprising:

processing information regarding an attempted telephone call;

determining if completion of said attempted telephone call meets operational objectives;

blocking said attempted telephone call if said attempted telephone call does not meet said operational objectives; and

generating an outbound call independent of said attempted telephone call to communicate an offer for facilitating future telephone calls if said attempted telephone call is blocked.

43. The method of claim **42,** wherein said attempted telephone call comprises a collect telephone call.

44. The method of claim **43,** wherein said collect telephone call is placed by an individual incarcerated in a prison facility.

45. The method of claim **42,** wherein said operational objectives comprises an acceptable level of risk regarding payment for carrying said telephone call.

46. The method of claim **42,** wherein said offer comprises presentation of one or more possible revenue opportunity applications.

47. The method of claim **42,** wherein said offer is communicated using an interactive voice response unit.

* * * * *

US005974114A

United States Patent [19]

Blum et al.

[11] **Patent Number: 5,974,114**

[45] **Date of Patent: Oct. 26, 1999**

[54] **METHOD AND APPARATUS FOR FAULT TOLERANT CALL PROCESSING**

[75] Inventors: **Andrea G. Blum**, Middletown; **Paul A. Potochniak**, Jackson, both of N.J.

[73] Assignee: **AT&T Corp**, New York, N.Y.

[21] Appl. No.: **08/937,762**

[22] Filed: **Sep. 25, 1997**

[51] **Int. Cl.**[6] **H04M 1/24**; H04M 7/00; H04M 3/00; G06F 11/00

[52] **U.S. Cl.** **379/9**; 379/221; 379/269; 370/217; 714/6; 714/17

[58] **Field of Search** 379/1, 9, 10, 14, 379/15, 32, 34, 219–221, 268, 269, 279; 370/216, 217, 220; 395/182.04, 181, 182.07, 182.08, 182.09; 714/5, 6, 10, 11

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,914,572	4/1990	Bitzinger et al.	379/10
4,949,373	8/1990	Baker, Jr. et al.	379/266
5,182,750	1/1993	Bales et al.	370/217
5,661,719	8/1997	Townsend et al.	370/216
5,848,128	12/1998	Frey	379/9
5,883,939	3/1999	Friedman et al.	379/9

OTHER PUBLICATIONS

IEEE publication 0018–9162/97, *Computer*, "Software–Based Replication for Fault Tolerance", Rachid Guerraoui and Andre Schiper, pp. 68–74.

Primary Examiner—Paul Loomis

[57] **ABSTRACT**

A method and apparatus for processing call data. A first server in active mode replicates call data to a second server in standby mode. The first server is monitored for a fault condition by the second server, as well as other network devices. If a fault condition is detected, the first server is switched to standby mode and the second server to active mode.

23 Claims, 7 Drawing Sheets

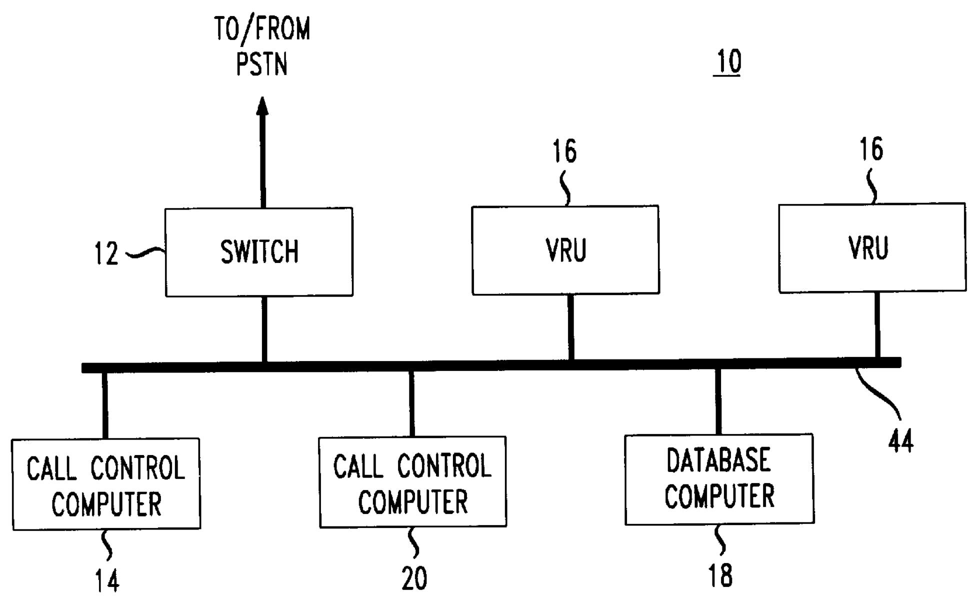


FIG. 1

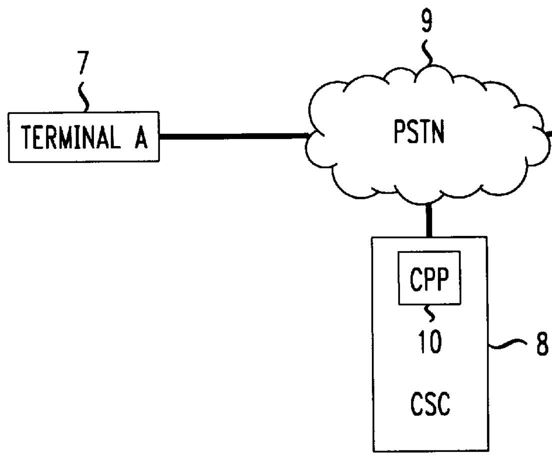

9
7
TERMINAL A
PSTN
7
TERMINAL B
CPP
10
8
CSC

FIG. 2

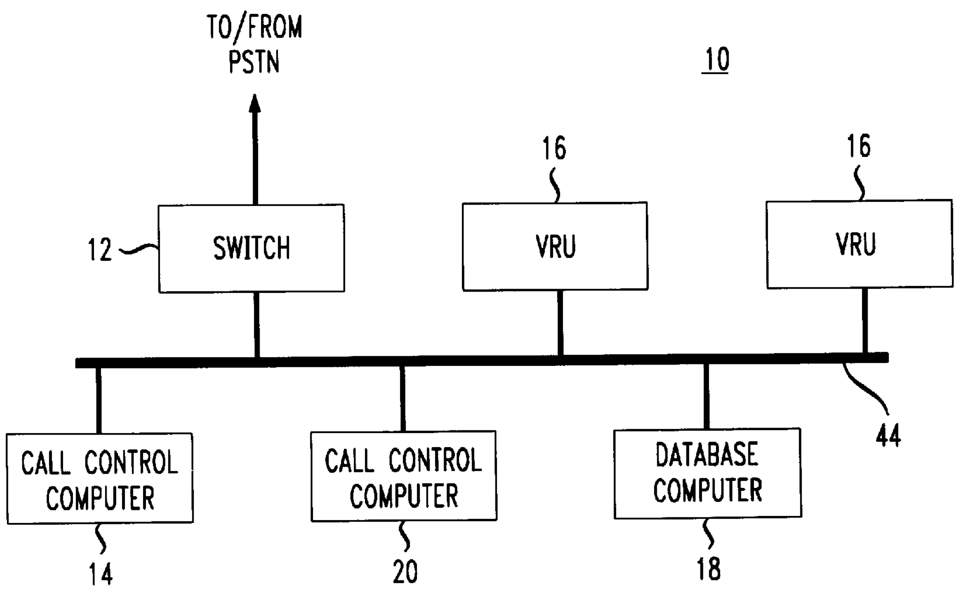

TO/FROM
PSTN
10
16
16
12
SWITCH
VRU
VRU
44
CALL CONTROL
COMPUTER
CALL CONTROL
COMPUTER
DATABASE
COMPUTER
14
20
18

FIG. 3

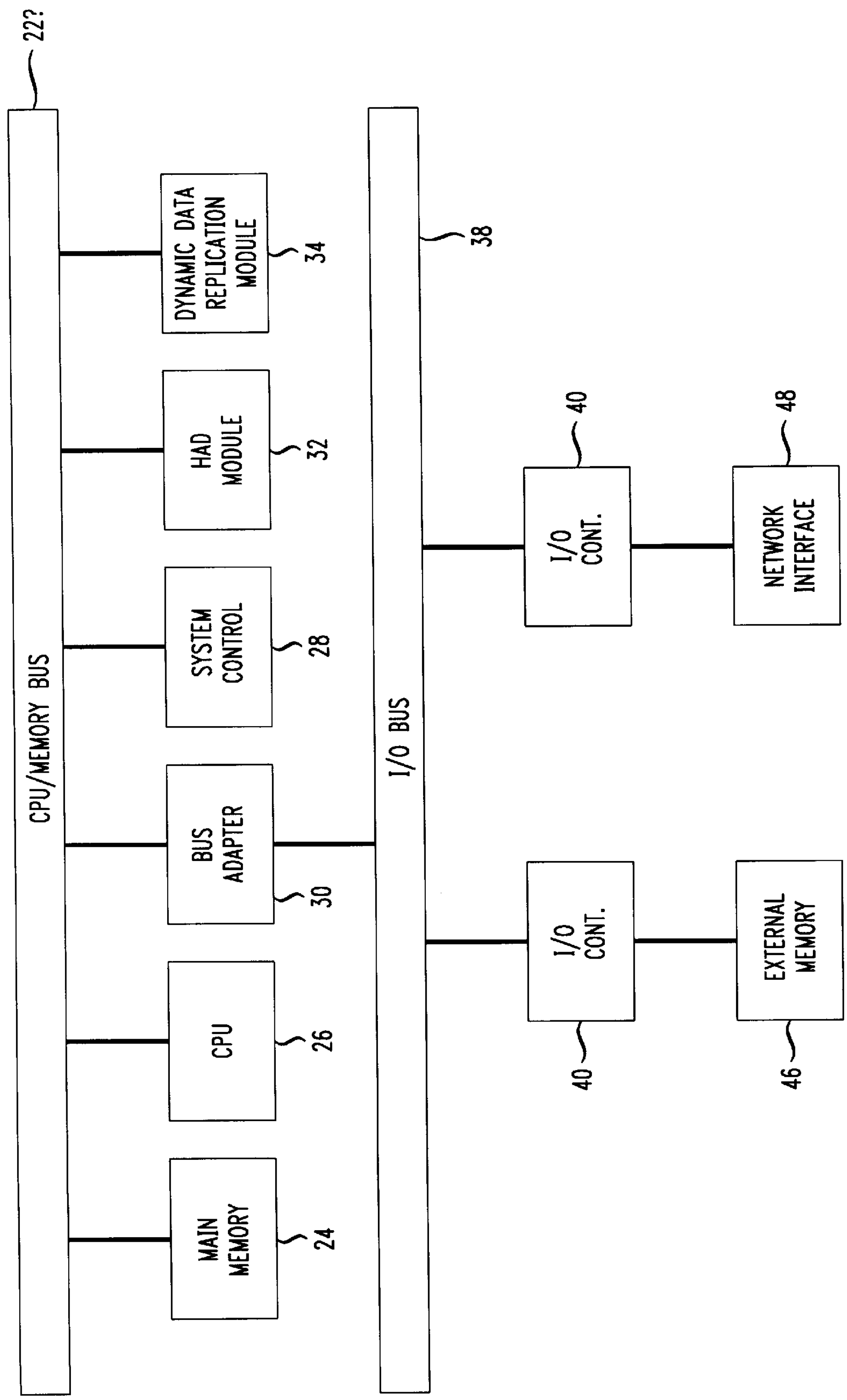
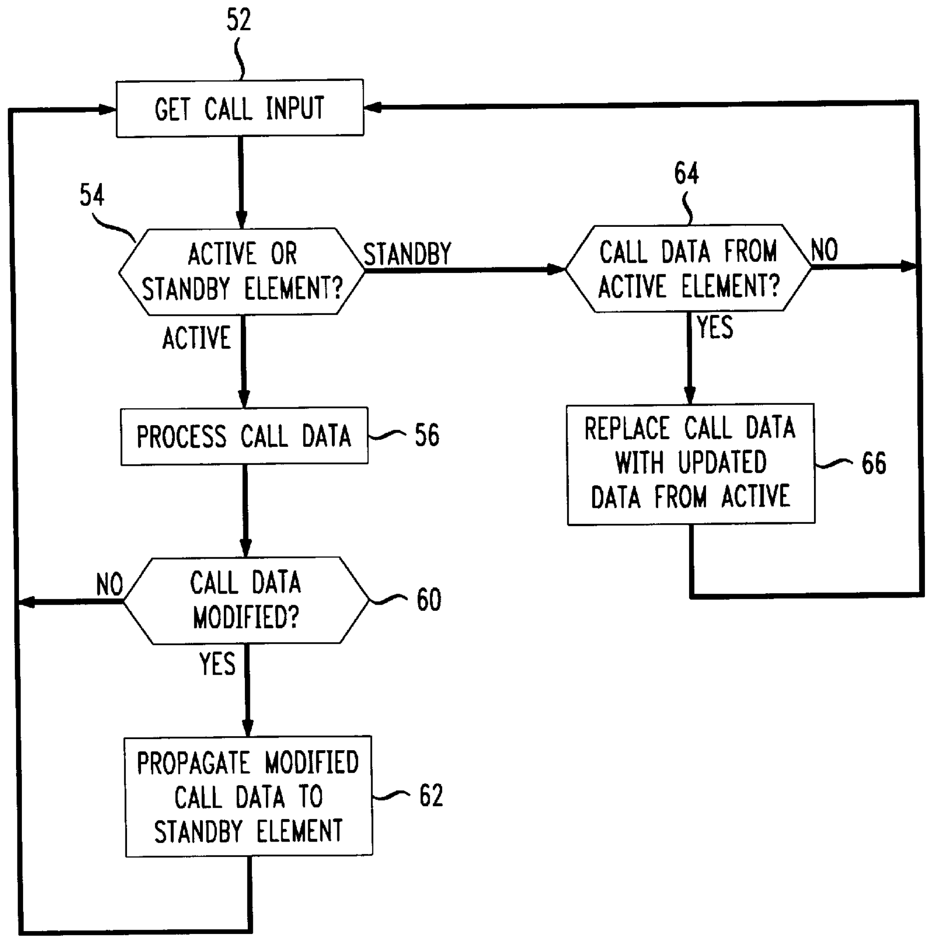

22?
CPU/MEMORY BUS
DYNAMIC DATA REPLICATION MODULE
34
HAD MODULE
32
SYSTEM CONTROL
28
BUS ADAPTER
30
CPU
26
MAIN MEMORY
24
38
I/O BUS
40
I/O CONT.
48
NETWORK INTERFACE
I/O CONT.
40
EXTERNAL MEMORY
46

FIG. 4



52
GET CALL INPUT
54
ACTIVE OR STANDBY ELEMENT?
STANDBY
ACTIVE
64
CALL DATA FROM ACTIVE ELEMENT?
NO
YES
PROCESS CALL DATA
56
REPLACE CALL DATA WITH UPDATED DATA FROM ACTIVE
66
NO
CALL DATA MODIFIED?
60
YES
PROPAGATE MODIFIED CALL DATA TO STANDBY ELEMENT
62

FIG. 5A

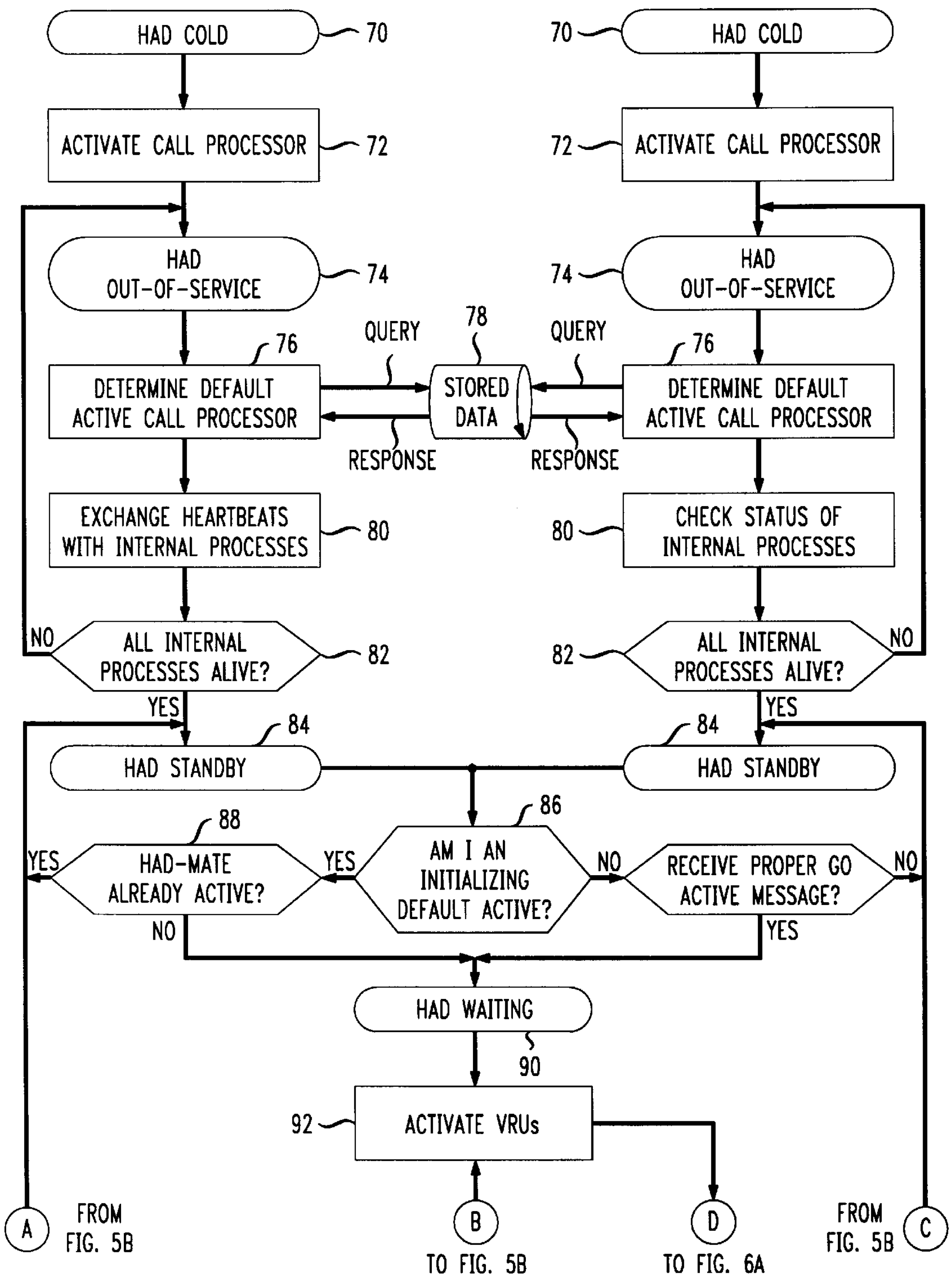
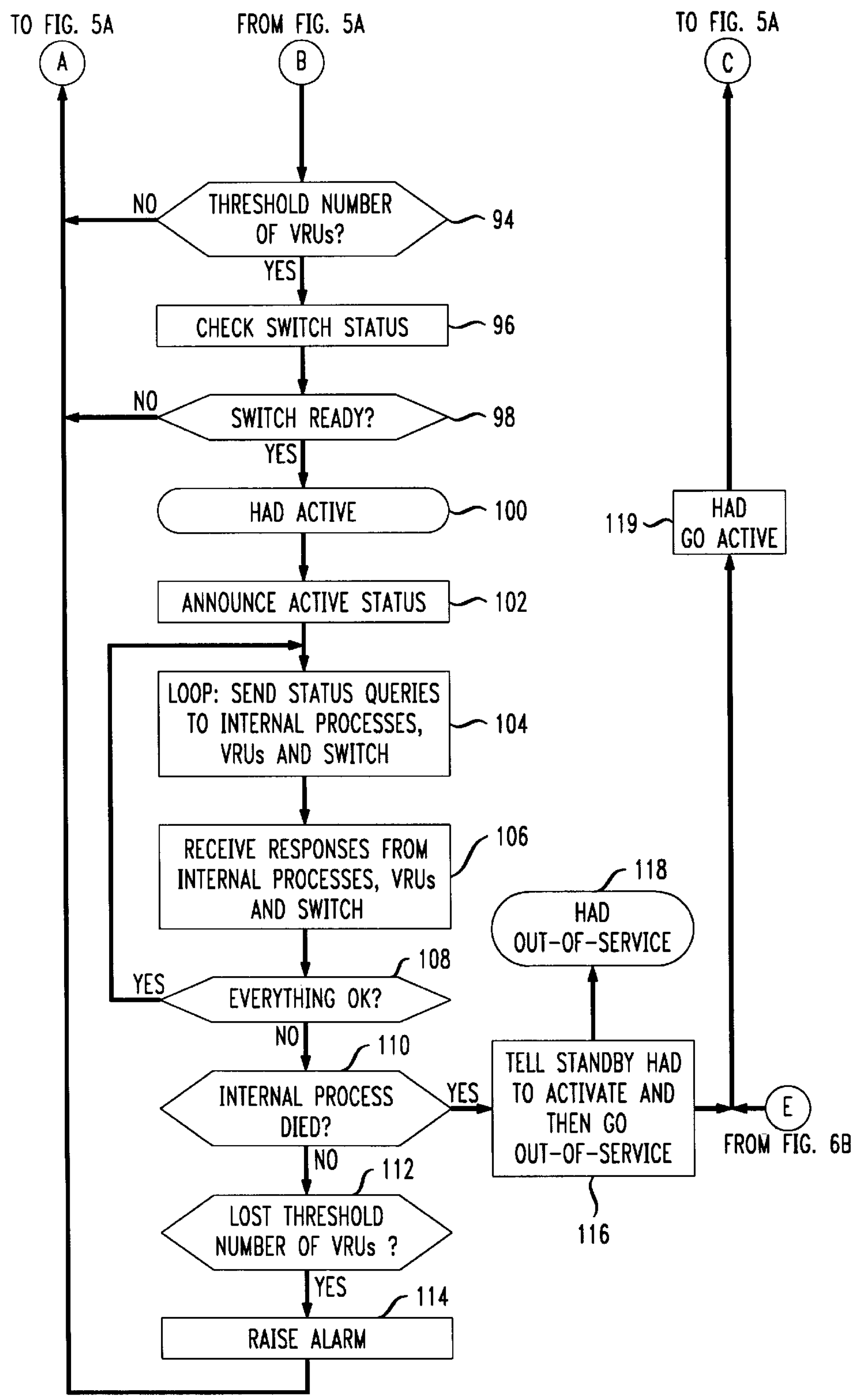

HAD COLD
70
ACTIVATE CALL PROCESSOR
72
HAD OUT-OF-SERVICE
74
DETERMINE DEFAULT ACTIVE CALL PROCESSOR
76
QUERY
RESPONSE
STORED DATA
78
EXCHANGE HEARTBEATS WITH INTERNAL PROCESSES
80
CHECK STATUS OF INTERNAL PROCESSES
ALL INTERNAL PROCESSES ALIVE?
82
NO
YES
HAD STANDBY
84
AM I AN INITIALIZING DEFAULT ACTIVE?
86
HAD-MATE ALREADY ACTIVE?
88
RECEIVE PROPER GO ACTIVE MESSAGE?
HAD WAITING
90
ACTIVATE VRUs
92
A
FROM FIG. 5B
B
TO FIG. 5B
D
TO FIG. 6A
C
FROM FIG. 5B

FIG. 5B



TO FIG. 5A
A
FROM FIG. 5A
B
TO FIG. 5A
C
NO
THRESHOLD NUMBER OF VRUs?
94
YES
CHECK SWITCH STATUS
96
NO
SWITCH READY?
98
YES
HAD ACTIVE
100
HAD GO ACTIVE
119
ANNOUNCE ACTIVE STATUS
102
LOOP: SEND STATUS QUERIES TO INTERNAL PROCESSES, VRUs AND SWITCH
104
RECEIVE RESPONSES FROM INTERNAL PROCESSES, VRUs AND SWITCH
106
118
HAD OUT-OF-SERVICE
YES
EVERYTHING OK?
108
NO
110
INTERNAL PROCESS DIED?
YES
TELL STANDBY HAD TO ACTIVATE AND THEN GO OUT-OF-SERVICE
E
FROM FIG. 6B
NO
112
LOST THRESHOLD NUMBER OF VRUs ?
116
YES
114
RAISE ALARM

FIG. 6A

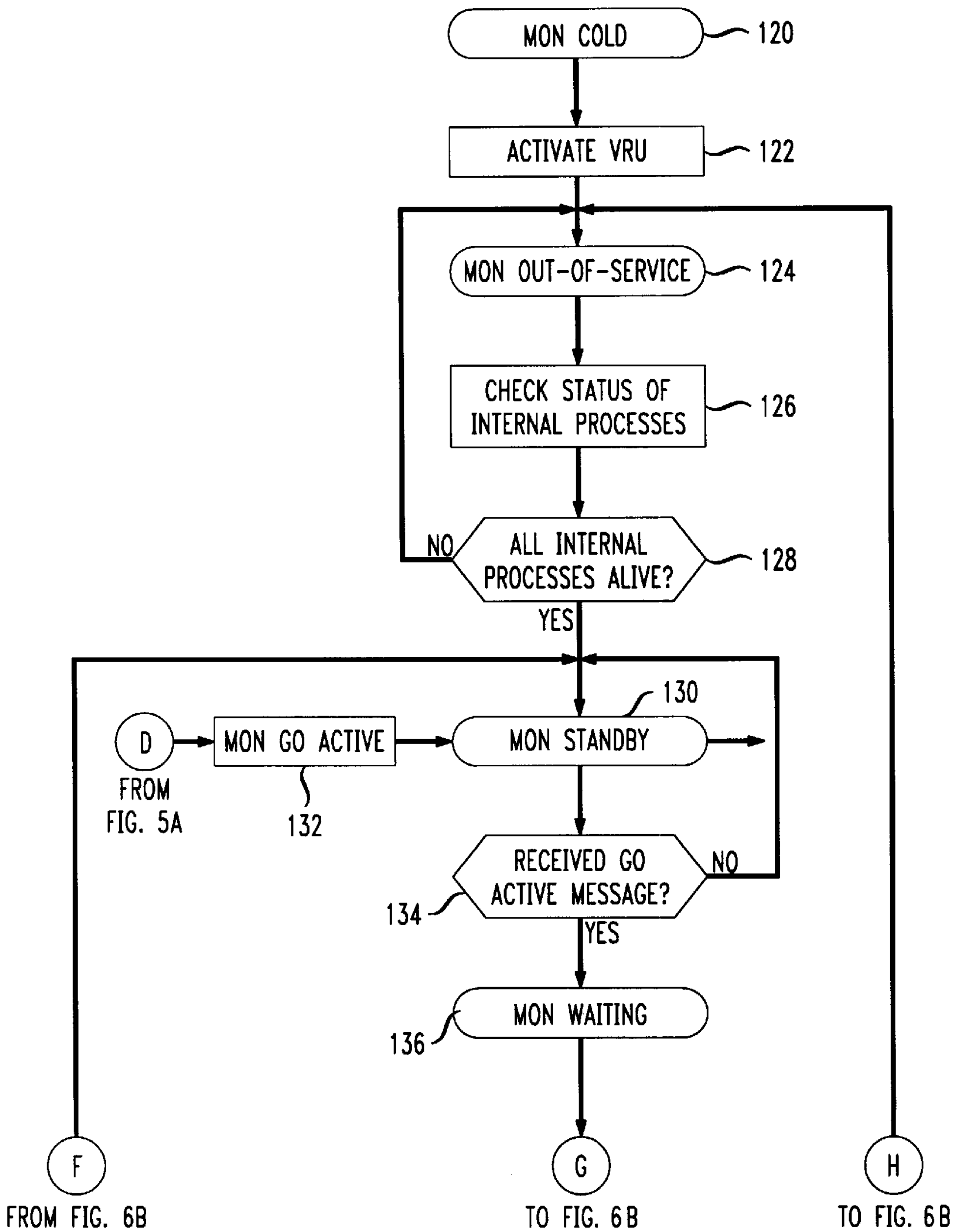
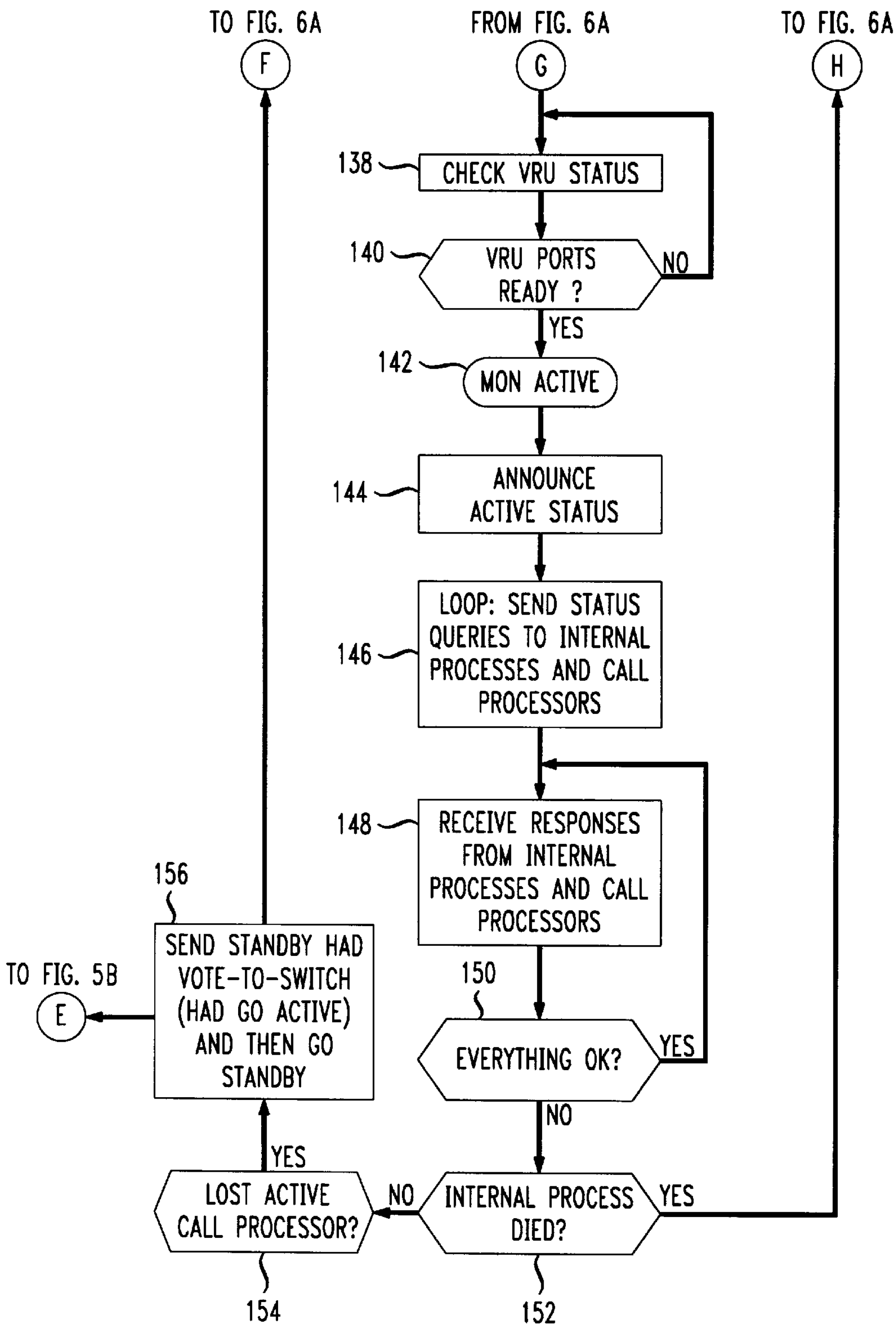

MON COLD
120
ACTIVATE VRU
122
MON OUT-OF-SERVICE
124
CHECK STATUS OF INTERNAL PROCESSES
126
ALL INTERNAL PROCESSES ALIVE?
128
NO
YES
130
D
FROM FIG. 5A
MON GO ACTIVE
132
MON STANDBY
RECEIVED GO ACTIVE MESSAGE?
134
NO
YES
MON WAITING
136
F
FROM FIG. 6B
G
TO FIG. 6B
H
TO FIG. 6B

FIG. 6B


TO FIG. 6A
F
FROM FIG. 6A
G
TO FIG. 6A
H
138
CHECK VRU STATUS
140
VRU PORTS READY ?
NO
YES
142
MON ACTIVE
144
ANNOUNCE ACTIVE STATUS
146
LOOP: SEND STATUS QUERIES TO INTERNAL PROCESSES AND CALL PROCESSORS
148
RECEIVE RESPONSES FROM INTERNAL PROCESSES AND CALL PROCESSORS
156
SEND STANDBY HAD VOTE-TO-SWITCH (HAD GO ACTIVE) AND THEN GO STANDBY
TO FIG. 5B
E
150
EVERYTHING OK?
YES
NO
YES
LOST ACTIVE CALL PROCESSOR?
NO
INTERNAL PROCESS DIED?
YES
154
152

METHOD AND APPARATUS FOR FAULT TOLERANT CALL PROCESSING

FIELD OF THE INVENTION

The invention relates to a call processing in general. More particularly, the invention relates to a method and apparatus for automatically switching call processing from an active call processor to a standby call processor in the event the active call processor fails.

BACKGROUND OF THE INVENTION

Given the current state of telephony technology, telephone calls over modern telecommunications networks are relatively reliable in terms of speed in completing a call connection, meeting quality of service requirements, and maintaining a call connection during the course of a conversation. The last category, maintaining a call connection, is provided in large part by building redundancy into the network, especially in the call processing platform. The call processing platform generally controls the set-up and shutdown of a call connection, and ensures that billing for a call is accurately maintained. This redundancy in the call processing platform ensures that a call connection is maintained even if there is a hardware failure in the equipment used to establish the call connection, and is sometimes referred to as "fault tolerant call processing."

Conventional technology and methods to build redundancy in a call processing platform, however, are less than desirable for a number of reasons. For example, a call processing platform typically has a call control computer that is responsible for implementing call flow by coordinating and assigning the resources of the other platform components, such as a switching matrix, voice response computers, and data base computers. Given its central function, the operation of the call control computer is extremely important in maintaining a call connection. Consequently, the call control computer is typically a specialized computer designed with redundant hardware components, such as a back-up microprocessor, memory, power supply, and so forth. This specialized call control computer, however, is very expensive. In addition, a single call control computer, even with redundant hardware, is susceptible to common mode failure. Common mode failure occurs when a single failure of a system component causes total system failure to occur. Further, the specialized call control computer is difficult to upgrade and maintain.

In an attempt to avoid the above problems, some call processing platforms utilize multiple call control computers, rather than a single dedicated call control computer with redundant hardware. The use of multiple call control computers, however, poses a new set of problems. Typically, one of the call control computers is designated as an active call control computer, with a second designated a standby call control computer. The active call control computer actively controls call processing functions for the call processing platform, while the standby call control computer stands ready to take over control of the call processing platform in the event the active call control computer experiences a hardware or software failure. To ensure that calls are not dropped when the active call control computer fails, it becomes necessary to duplicate all call processing data to the standby call control computer. Further, it becomes necessary to implement a monitoring scheme to monitor the active call control computer, and determine when it becomes necessary to switch over to the standby call controller.

Conventional techniques exist for duplicating call processing data from an active call control computer to a standby call control computer, such as the technique disclosed in a paper authored by Rachid Guerraoui et al. Titled "Software Based Replication for Fault Tolerance," Computer Journal, IEEE, April 1997. The technique described in the Guerraoui paper, however, is unsatisfactory for a number of reasons. For example, the Guerraoui paper fails to disclose a monitoring and switch over scheme that minimizes dropped calls in the case of failure of the active call control computer. Further, the Guerraoui paper fails to disclose a means for synchronizing the call processing data across the call processing platform.

In addition, the Guerraoui paper fails to teach how to ensure that the standby computer has accurate records regarding static call data. Typically, a call processing platform requires two types of data to process a call: (1) dynamic call data; and (2) static call data. Dynamic call data is information about the caller or call connection that changes for every call. For example, a destination telephone number is considered dynamic call data since it typically changes from call to call. Static call data is information about a caller that is relative stable, that is, it does not change on a call by call basis. An example of static call data would be a billing address for a caller, or perhaps a Personal Identification Number. The Guerraoui paper fails to discuss the duplication of static call data to the standby call control computer.

In view of the foregoing, it can be appreciated that a substantial needs exists for a fault tolerant call processing method and apparatus that solves the above-discussed problems.

SUMMARY OF THE INVENTION

The present invention includes method and apparatus for processing call data. A first server in an active mode replicates call data to a second server in a standby mode. The first server is monitored for a fault condition by the second server, as well as other network devices. If a fault condition is detected, the first server is switched to standby mode and the second server to active mode.

With these and other advantages and features of the invention that will become hereinafter apparent, the nature of the invention may be more clearly understood by reference to the following detailed description of the invention, the appended claims and to the several drawings attached herein.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates a communications system suitable for practicing one embodiment of the invention.

FIG. **2** illustrates a call processing platform in accordance with one embodiment of the invention.

FIG. **3** is a block diagram of a call control computer in accordance with one embodiment of the invention.

FIG. **4** illustrates a block flow diagram of steps performed by a dynamic data replication module in accordance with one embodiment of the invention.

FIG. **5**(*a*) illustrates a first block flow diagram of a High Availability Daemon (HAD) module in accordance with one embodiment of the invention.

FIG. **5**(*b*) illustrates a second block flow diagram of a HAD module in accordance with one embodiment of the invention.

FIG. **6**(*a*) illustrates a first block flow diagram of a Monitor Service (MON) module in accordance with one embodiment of the invention.

FIG. **6**(*b*) illustrates a second block flow diagram of a MON module in accordance with one embodiment of the invention.

DETAILED DESCRIPTION

The invention includes a method and apparatus for fault tolerant call processing. More particularly, the invention includes a method and apparatus for automatically switching from an active call control computer to a standby call control computer in the event of a hardware or software failure of the active call control computer, without interrupting active call connections being processed by the active call control computer. Two key elements required to perform this automatic switch over are call data synchronization and communications monitoring.

One embodiment of the invention comprises a call processing platform built upon general purpose computer devices. The general purpose, non-specialized computing devices are combined with voice response units (VRUs) and a switching matrix to create a distributed, fault tolerant, easily maintained call processing platform that provides high service availability through the use of "hot" standby sparing, full data sharing, database replication and synchronization, and a software-based distributed monitoring system. It is worthy to note that although the distributed monitoring system of this embodiment of the invention is implemented in software, it can be appreciated that the distributed monitoring system could be implemented in hardware or software and still fall within the scope of the invention.

The call processing platform performs call control and resource management using general purpose, non-specialized computer devices. The use of general purpose, non-specialized computer devices significantly reduces the cost of the call processing platform in general, and the call control computers in particular. This embodiment of the invention utilizes a pair of general purpose, non-specialized computer devices as call control computers, with one of the computers actively controlling call processing for the call processing platform ("active call control computer"), and with the other placed in a standby mode ("standby call control computer") and ready to assume call processing responsibilities in the event the active call control computer experiences a hardware or software failure.

Switching from the active call control computer to the standby call control computer can be performed on demand or automatically in the event of failure of the active call control computer. The on-demand "active/standby switch over" of the call control computers permits a platform administrator to request either an ON_DEMAND GRACEFUL switch over or an ON_DEMAND QUICK switchover. The ON_DEMAND GRACEFUL switch over resynchronizes the entire call processing platform by temporarily halting call processing and cleaning up all currently utilized switch resources. The ON_DEMAND QUICK switch over operates similar to the automatic active/standby switch over described below.

The automatic active/standby switch over of the call control computers is accomplished utilizing two key elements. The first key element is platform monitoring. The second key element is synchronizing call state information.

Platform monitoring is accomplished using distributed monitors for the call control computers and other critical processes. Each call control computer is equipped with a communications monitor for monitoring the internal processes for the call control computer, as well as the health of the other network devices that are part of the call processing platform. In addition, each network device is equipped with a communications monitor for monitoring the internal processes of each network device, as well as the call control computers. Each communications monitor can detect failure of the device that is running the monitor, as well as the failure of other devices external to the device running the monitor. Thus, each network device, including the call control computers, is capable of detecting device failures and reporting the device failures to the active call control computer. Additionally, each communications monitor remote to the active call computer can detect or confirm communication failure of the active call control computer and alert the standby call control computer of the need for a possible takeover.

In this embodiment of the invention, platform monitoring is accomplished through the use of two sets of monitoring processes. These processes monitor the platform for hardware and software failure so that call processing is maintained by activating the standby call control computer upon the failure of the active call control computer.

The first set of monitoring processes are referred to as High Availability Daemon (HAD) processes. The HAD processes run on the call control computers, with one HAD per computer. The HADs are responsible for: (1) coordinating startup and shutdown of call processing on the platform; (2) tracking the health of applications local to their own processors; (3) tracking the communication status and system state of the other platform components; and (4) monitoring the health of each other's call control computer. The HAD process is described in more detail with reference to FIGS. **3**, **5**(*a*) and **5**(*b*).

The second set of monitoring processes are referred to as Monitor Service (MON) processes. The MON processes runs on the other components of the platform, e.g., VRUs and database computer. Each component has one MON process. In general, MONs are responsible for: (1) tracking the health of the application local to their own processor; (2) reporting the state of the local processor to the two call control computers; and (3) directing call flow to the active call control computer. The MON process is described in more detail with reference to FIGS. **3**, **6**(*a*) and **6**(*b*).

If any of the monitoring processes (HADs or MONs) detects a failure that affects the call processing capabilities of the active call control computer, they register a vote-to-switch with the standby call control computer. Upon receiving two such votes the standby activates. First the standby tells its (formerly active) mate call control computer to enter a standby mode. Then the standby informs the other platform components to redirect the call flow to it as the newly active call control computer.

The other key component of the automatic active/standby switch over is the fully associated synchronization of each call state data structure contained on the active call control computer with its replicated call state data structure on the standby call control computer. As part of normal operation the call control computer maintains call information on a per call basis, i.e., dynamic call data. This information deals with switch and VRU resources currently assigned to a call, and caller data such as a target number and billing instrument (e.g., calling card) data. As this information is collected by the active call control computer from the other platform elements, the data is synchronized in real time to the standby call control computer. By this method, the standby call control computer always has all call information necessary to continue call processing should the monitoring processes determine that the active has failed.

Thus, the call control computers are fully synchronized with respect to call data used for the call processing. The active call control computer immediately shares all call state updates with the hot standby so that upon the active's failure, the standby can accept re-directed call flow with minimal loss of active calls or queuing delay.

Database synchronization and replication of static call data is also performed for both call control computers. a database computer stores static call data in a static call data profile, and then replicates the static call data onto both the active and standby call control computers whenever the static call data is accessed or modified. This ensures that should data be lost on any unit, it may be easily recovered from a replication. The replication of static call data for this embodiment of the invention utilizes an Advanced Replication product provided by Oracle Corporation. The call server copies of the database are read-only, and propagated to the call servers using Oracle's Read-Only Snapshots product.

Periodic data audits of dynamic and static call data records on both call control computers are performed to confirm that all data is synchronized. This ensures that both call control computers have updated call records regarding a particular call so that the call is not dropped in the event of a failure by the active call control computer.

Referring now in detail to the drawings wherein like parts are designated by like reference numerals throughout, there is illustrated in FIG. **1** a communications system suitable for practicing one embodiment of the invention. As shown in FIG. **1**, terminals A and B (each labeled number **7**) are connected to a Public Switched Telephone Network (PSTN) **9**. PSTN **9** is also connected to a Call Servicing Center (CSC) **8**. A calling party initiates a telephone call from terminal A. The call is processed by CSC **8**, and a call connection is completed to the called party at terminal B via PSTN **9**. CSC **8** includes a call processing platform (CPP) **10** that is described in more detail with reference to FIG. **2**.

FIG. **2** illustrates a call processing platform in accordance with one embodiment of the invention. A CPP **10** includes a computer controlled switching matrix **12**, a first call control computer **14**, a second call control computer **20**, a plurality of VRUs **16**, and a database computer **18**.

Switching matrix **12** interfaces with a pair of call control computers via local area network (LAN) **44**. Switching matrix **12** is responsible for providing all network terminations to the PSTN.

Call control computers **14** and **20** are responsible for the implementation of call flow between an origination number and a destination number. Call control computers **14** and **20** coordinate and assign the resources of the other platform components such as switch **12**, VRUs **16** and database computer **18**. Each call control computer has an active mode and a standby mode. A call control computer in active mode actively controls call processing for CPP **12**, while the other call computer is placed in standby mode as a back-up to the call control computer in active mode.

VRUs **16** are computers capable of providing speech and touch tone resources used to interact with the caller. VRUs **16** are connected to switching matrix **12** via a network such as an Integrated Services Digital Network Primary Rate Interface (ISDN-PRI), and to call control computer **14** over another network, such as LAN **44**.

Database computer **18** is a general purpose computer containing a relational database for use in call processing. Database computer **18** is connected to the call control computers via LAN **44**.

FIG. **3** is a block diagram of a call control computer in accordance with one embodiment of the invention. For purposes of clarity, the following description will make reference to call computer **14**. Call control computers **14** and **20** are similar, however, and therefore any discussion regarding one call control computer is equally applicable to the other call control computer.

Call control computer **14** comprises a main memory module **24**, a central processing unit (CPU) **26**, a system control module **28**, a bus adapter **30**, a High Availability Daemon (HAD) module **32**, and a dynamic data replication module **34** each of which is connected to a CPU/memory bus **22** and an Input/Output (I/O) bus **38** via bus adapter **30**. Further, call control computer **20** contains multiple I/O controllers **40**, as well as a external memory **46** and a network interface **48**, each of which is connected to I/O bus **38** via I/O controllers **40**.

The overall functioning of call control computer **14** is controlled by CPU **26**, which operates under the control of executed computer program instructions that are stored in main memory **24** or external memory **46**. Both main memory **24** and external memory **46** are machine readable storage devices. The difference between main memory **24** and external memory **46** is that CPU **26** can typically access information stored in main memory **24** faster than information stored in external memory **36**. Thus, for example, main memory **24** may be any type of machine readable storage device, such as random access memory (RAM), read only memory (ROM), programmable read only memory (PROM), erasable programmable read only memory (EPROM), electronically erasable programmable read only memory (EEPROM). External memory **46** may be any type of machine readable storage device, such as magnetic storage media (i.e., a magnetic disk), or optical storage media (i.e., a CD-ROM). Further, call control computer **14** may contain various combinations of machine readable storage devices through other I/O controllers, which are accessible by CPU **26**, and which are capable of storing a combination of computer program instructions and data.

CPU **26** includes any processor of sufficient processing power to perform the HAD and data replication functionality found in call control computer **14**. Examples of CPUs suitable to practice the invention includes the INTEL family of processors, such as the Pentium®, Pentium® Pro, and Pentium® II microprocessors.

Network interface **48** is used for communications between call control computer **14** and a communications network, such as LAN **44**. Network interface **48** supports appropriate signaling and voltage levels, in accordance with techniques well known in the art.

I/O controllers **40** are used to control the flow of information between call control computer **14** and a number of devices or networks such as external memory **46** and network interface **48**. System control module **28** includes human user system control, user interface, and operation. Bus adapter **30** is used for transferring data back and forth between CPU/memory bus **22** and I/O bus **38**.

VRUs **16** and database computer **18** are similar to call control computer **14** described with reference to FIG. **3**. VRUs **16** and database computer **18**, however, replaces HAD module **32** with a Monitor Service (MON) module **50** (not shown in FIG. **3**). MON **50** may also be implemented on other network devices internal or external to CPP **10**.

HAD **32**, MON **50** and dynamic data replication module **34** implements the main functionality for this embodiment of the invention. It is noted that HAD module **32** and dynamic data replication module **34** are shown in FIG. **3** as, and MON module **50** is described as, separate functional

modules. It can be appreciated, however, that the functions performed by these modules can be further separated into more modules, combined together to form one module, or be distributed throughout the system, and still fall within the scope of the invention. Further, the functionality of these modules may be implemented in hardware, software, or a combination of hardware and software, using well-known signal processing techniques.

HAD **32** and MON **50** share responsibility for four central functions: (1) coordinating startup and shutdown of call control computers **14** and **20**; (2) tracking and logging communication and activity states for call control computer **14** and **20**; (3) detecting and alarming any hardware, software or other failures/problems; and (4) monitoring the operations of each other.

HAD **32** runs on both call control computers **14** and **20**. Call control computers **14** and **20** have two primary modes: (1) an active mode; and (2) a standby mode. When a call control computer is in active mode, it is actively controlling call processing functions for CPP **10**, and is referred to as an active call control computer. Similarly, HAD **32** running on the active call control computer is referred to as an active HAD (HAD-CurrActy). When a call control computer is in standby mode, it is kept ready to take over active control of the call processing functions for CPP **10** either on-demand or automatically with minimal impact on currently active calls. A call control computer in standby mode is referred to as a standby call control computer, and HAD **32** running on the standby call control computer is referred to as a standby HAD (HAD-Stand). At any time, only one of the two call control computers may be in active control of CPP **10**.

HAD **32** provides the following functionality for call control computers **14** and **20**:

1. Bringing up and shutting down the active call control computer's critical processes in a particular order during platform startup and shutdown.

2. Notifying the MONs running on the other network devices to bring up or shut down critical processes on the other network devices.

3. Performing on-demand or automatic switch-over of platform control between call control computers **14** and **20**.

4. The standby HAD recognizes the need for, and initiates, automatic switch-over of platform control to the standby call control computer from a failed active call control computer with minimal loss of currently active calls.

5. Keeping track of the status of a call server's critical processes.

6. Keeping track of the status of other network devices' critical processes.

7. Recognizing which is the default active call control computer upon cold-start or re-start and automatically initializing the default active call control computer accordingly.

8. Responding to any MON's heartbeats or state queries from other network devices.

MON **50** runs on all network devices remote to call control computers **14** and **20**, such as VRUs **16** and database computer **18**. MON **50** provides the following functionality for these other network devices:

1. Recognizing which is the currently active call control computer by communication with the HAD-CurrActy.

2. Responding to either HAD's heartbeats, state queries, state change reports and state transition requests.

3. Keeping track of the status of the other network devices' critical processes.

4. Notifying the currently active HAD of any state changes or alarms.

5. Monitoring the communication status of the HAD-CurrActy and notifying the standby HAD of any problems.

To properly implement automatic switch over, the HAD **32** or MON **50** processes must detect and act upon system failures within a short period of time, e.g., 5 seconds of their occurrence. The type of failures that may be detected by HAD **32** or MON **50** include:

1. The failure of a critical process on a call server;

2. The loss of heart beat messages from a critical process; or

3. The loss of the active call server due to network or operating system failure.

Additional details for HAD **32** and MON **50** will be described later in this specification.

Dynamic data replication module **34** is responsible for replicating call data received at the active call control computer to the standby call control computer. Thus, if the active call control computer fails, the standby call control computer can take over call processing operations for CPP **10** while minimizing the number of calls dropped during the switch over process. Dynamic data replication module **34** is described in more detail with reference to FIG. **4**.

FIG. **4** illustrates a block flow diagram of steps performed by a dynamic data replication module in accordance with one embodiment of the invention. As shown in FIG. **4**, call data is received at step **52**. At step **54**, the system determines whether the active call control computer or standby call control computer is to receive the call data.

If the active call control computer is to receive the call data at step **54**, the active call control computer processes the call data at step **56**. Active call control computer accesses a call data record, and compares the received call data with the call data stored in the call data record at step **60**. If the call data differs from the call data stored in the call data record at step **60**, the call data is replicated and sent to the standby call control computer at step **62**. If the call data is not different from the call data stored in the call data record at step **60**, the system looks for the next set of call data at step **52**.

If the standby call control computer is to receive the call data at step **54**, the system determines whether the call data is from the active call control computer at step **64**. If it is, the call data record for the standby call control computer is updated with the new call data at step **66**. If the call data is not from the active call control computer at step **64**, the system looks for the next set of call data at step **52**.

Database computer **18** is a general purpose computer containing a relational database for use in call processing. As with the other network devices described with reference to CPP **10**, database computer **18** includes a MON module **50** for monitoring call control computers **14** and **20**. Database computer **18** also includes a static data replication module. The static data replication module receives static call data, and stores the static call data in a static call data profile in the relational database. Every time the static call data profile is updated, the static data replication module replicates the static call data stored in the static call data profile to call control computers **14** and **20**.

CPP **10** periodically audits the call data records and the static call data profiles on a periodic basis. The data audits help ensure data synchronization between call control computers **14** and **20**.

FIG. **5**(*a*) illustrates a first block flow diagram of a High Availability Daemon (HAD) module in accordance with one

embodiment of the invention. CPP **10** has two call control computers, a first call control computer and a second call control computer. Each call control computer executes a HAD process, with each HAD process in communication with the other. For purposes of clarity, a HAD process running on the first call control computer will be referred to as "the first HAD process," and a HAD process running on the second call control computer will be referred to as "the second HAD process." Similarly, a HAD process running on the active call control computer will be referred to as "the active HAD process" and a HAD process running on the standby call control computer will be referred to as "the standby HAD process."

As shown in FIG. **5**(*a*), each HAD process executes steps **70**, **72**, **74**, **76**, **78**, **80**, **82** and **84**. At step **70**, the HAD process is initiated. Upon start up, the HAD process activates the call control computer on which it is running at step **72**. At step **74**, the HAD is taken out of service. At Step **76**, the HAD process determines whether the call control computer upon which it is running is the default active call control computer. In this embodiment of the invention, this determination is accomplished by querying stored data at step **78** and receiving a response to the query at step **76**. Alternatively, other means could be implemented for choosing the default active call processor, such as through an alternating or random selection process, and still fall within the scope of the invention. At step **80**, the HAD process exchanges heart beats with the internal processes running on the same call control computer that is running the HAD process. The HAD process determines whether all the internal processes are operating within normal performance parameters at step **82**. If at step **82** all the internal processes are not operating according to normal performance parameters, then the HAD process is placed out of service again at step **71**. If all internal processes are operating according to normal performance parameters at step **82**, the HAD process is put in standby mode at step **84**.

Thus at step **84**, both HAD processes are placed in standby mode. The default active HAD process is initialized at step **86**. The default active HAD processor then determines whether the other HAD process ("HAD mate") is already in active mode at step **88**. If the HAD Mate is already active at step **88**, the default active HAD process is placed on standby at step **84**. If the HAD Mate is not already active at step **88**, then the default active HAD is placed into a waiting mode at step **90**. At step **92**, the default active HAD process activates VRU **16**, and sends a MON go active message **132** to MON **50**.

FIG. **5**(*b*) illustrates a second block flow diagram of a HAD module in accordance with one embodiment of the invention. The default active HAD process determines at step **94** whether the threshold number of VRU's have been activated. If the threshold number of VRU's have not been activated at step **94**, then the default active HAD is placed in standby mode at step **84**. If the threshold number of VRU's have been activated at step **94**, the default active HAD process checks the switch status at step **96**. The default active HAD process determines whether the switch is ready to perform switching functions at step **98**. If the switch is not ready at step **98**, the default active HAD process is put in standby mode at step **84**. If the switch is ready to perform switching functions at step **98**, then the default active HAD is placed in an active mode at step **100**. Once the active HAD has been placed in active mode, the HAD process announces its active status to all the other network devices at step **102**.

One function shared by the HAD processes and MON processes is to monitor the internal processes of the computer running the HAD or MON process, and also monitor network devices external to the computer running the HAD or MON process. At steps **104**, **106** and **108**, the active HAD process queries the internal processes of the active call control computer, as well as other network devices such as VRU **16** and Switching Matrix **12**. At step **104**, the active HAD process sends status queries to the internal processes, VRU **16** and Switching Matrix **12**. The HAD process receives responses from the internal processes, VRU **16** and Switching Matrix **12** at step **106**. At step **108**, the HAD process determines whether the internal processes, VRU **16** and Switching Matrix **12** are functioning properly. If the internal processes, VRU **16** and Switching Matrix **12** are operating properly at step **108**, then steps **104**, **106** and **108** are repeated until the HAD process determines that one of the internal processes, VRU **16** or Switching Matrix **12** is not operating appropriately at step **108**.

If a failure does occur in the internal processes, VRU **16** or Switching Matrix **12** at step **108**, the HAD process determines whether it is the internal processes that have failed at step **110**. If the internal processes have not failed at step **110**, the HAD process determines whether the call processing platform **10** has lost a threshold number of VRU **16** at step **112**. If a threshold number of VRUs are not present at step **112**, then an alarm is raised at step **114** and the active HAD process is placed on standby at step **84**.

If the active HAD process determines that an internal process has failed at step **110**, the active HAD process notifies the standby HAD process to activate, and then orders the active call control computer to go out of service at step **116**. The active HAD process is then placed out of service at step **118**.

FIG. **6**(*a*) illustrates a first block flow diagram of a Monitor Service (MON) module in accordance with one embodiment of the invention. FIG. **6**(*a*) shows a Mon process which may be running on any of the network devices that are part of CPP **10**. At step **120**, a Mon process is started. At step **122**, the Mon process activate, VRU **16**. At step **124**, the Mon process is placed out of service. The Mon process then checks the status of the internal processes of the device which is running the Mon process at step **126**. The Mon process determines whether all the internal processes are running properly at step **128**. If all the internal processes are not running properly at step **128**, then the Mon process is placed out of service at step **124**. If, however, all internal processes are running properly at step **128**, then the Mon process is placed in a standby mode at step **130**.

At step **134**, the MON process determines whether it has received a MON go active message **132**. If it has not received a MON go active message **132**, then the MON process remains on standby mode at step **130**. If a MON go active message **132** is received at step **130**, the Mon process is placed in a waiting mode at step **136**.

FIG. **6**(*b*) illustrates a second block flow diagram of a MON module in accordance with one embodiment of the invention. At step **138**, the Mon process checks the status of VRU **16**. At step **140**, the Mon process determines whether VRU **16** ports are ready. If the VRU ports are not ready at step **140**, then steps **138** and **140** are repeated until the VRU ports are ready. If the VRU ports are ready at step **140**, then the Mon process is placed in active mode at step **142**. The active Mon process announces its active status to the other network devices at step **144**.

Steps **146**, **148** and **150** perform the monitoring process for the active Mon process. At step **146**, the active Mon process sends status queries to the internal processes running on the device that is running the active Mon process, and

also checks the status of the active and standby call control computers. Responses from the internal processes and the active and standby call control computers are received at step **148**. The active Mon process determines at step **150** whether the internal processes and the active and standby call control computers are operating within normal performance parameters. If the internal processes and active and standby call control computers are working within normal parameters at step **150**, steps **148** and **150** are repeated until the Mon process determines that the internal processes, the active call control computer or the standby call control computer has failed at step **150**. In the event of a failure at step **150**, the Mon process determines whether an internal process has failed at step **152**. If an internal process has failed at step **152**, then the Mon process is placed out of service at step **124**. If an internal process has not failed at step **152**, then the active Mon process determines whether the active call processor has failed at step **154**. If the active call processor has failed at step **154**, the active Mon process sends the standby HAD process a vote-to-switch (HAD Go Active message **119**) at step **156**, and then goes into a standby mode at step **130**.

Referring again to FIG. **5**(*b*), a HAD Go Active message **119** is sent to the standby HAD process at step **84**. At step **86**, the standby HAD process determines whether it is being initialized as the default active HAD process. Since the standby HAD process is not being initialized as the default active HAD process at step **86**, the standby HAD process determines whether it has received a proper HAD go active message **119** at step **160**. If the standby HAD process has not received a proper HAD go active message **119** at step **160**, the standby HAD process remains in standby mode at step **84**. If the standby HAD process receives a proper HAD go active message **119** at step **160**, the standby HAD process performs steps **90** to **118** as the newly active HAD process.

The operation of CPP **10** can be better understood through the following example. Assume that a passenger on an airplane desires to make a telephone call. The passenger takes the handset portion of an air terminal off hook and presses an ON button. When the handset is turned ON, the air terminal seizes a radio channel to a ground station. The ground station performs a network connection to CPP **10**. A message is sent to the active call control computer of CPP **10** that Switching Matrix **12** has detected a trunk seizure. The active call control computer then begins a new call record, and replicates the new call record to the standby call control computer.

Switching Matrix **12** then notifies VRU **16** of an incoming call request. VRU **16** performs answer supervision and requests identification information from the ground station. The ground station sends back ground station/air terminal (GS/AT) identifiers to Switching Matrix **12**, which passes the GS/AT identifiers to VRU **16**. VRU **16** sends the GS/AT identifiers to the active call control computer. The active call control computer updates its call data record, and replicates the GS/AT identifiers to the standby call control computer so that it may update its call data record. The active call control computer then accesses a static call data profile on database computer **18** to validate the GS/AT identifiers. If the GS/AT identifiers are valid, the active call control computer updates its call data record and replicates the validated GS/AT identifiers to the standby call control computer so that it may update its call data record.

The active call control computer then sends a message to VRU **16** that it is clear to collect call information. VRU **16** passes the same message to Switching Matrix **12**. Switching Matrix **12** sends an acknowledgment to the ground station that the GS/AT identifiers are valid. The air terminal cuts a voice path to ground station, which in turn cuts a voice path to Switching Matrix **12**. Switching Matrix **12** then cuts a voice path to VRU **16**. VRU **16** plays a dial tone that is sent to the air terminal. Once the passenger on the airplane receives the dial tone, the passenger is prompted to swipe his or her credit card to pay for the telephone call. The credit card information is received by VRU **16**, which passes it along to the active call control computer to update its call data record. The active call control computer then replicates the credit card information to the standby call control computer so that the standby call control computer may update its call data record. The active processor then checks the static call data profile to determine whether the credit card number is a valid number. If the credit card number is valid, the active processor sends a message to VRU **16** that it is clear to collect a destination number from the passenger. The active call control computer also replicates the validation message to the standby call control computer. Once VRU **16** receives clearance to collect a destination number, VRU **16** plays the dial tone again for the passenger using the air terminal. The passenger enters a destination telephone number to complete a call connection. The destination telephone number is sent from the air terminal to VRU **16**, which in turn passes it to the active call control computer so that it may update its call record. The active call control computer then replicates the destination telephone number to the standby call control computer. As with the GS/AT identifiers and credit card information, the active call control computer validates the destination telephone number by accessing the static call data profile stored by Database Computer **18**. If the destination phone number is a valid destination phone number, the active call control computer sends a message to VRU **16** that it is okay to brand the call. The active call control computer also updates its own call record and replicates the validation of the destination telephone number to the standby call control computer so that it may update its own call data record. VRU **16** sends a message to the passenger indicating that the call connection has been made by sending a message such as "Thank you for using AT&T."

At this point, VRU **16** sends a message to the active call control computer that the call branding is completed. The active call control computer sends a message to Switching Matrix **12** to set up a communication link to the called party. Switching Matrix **12** initiates a communication link and waits for an answer from the called party. If switching matrix **12** receives an answer, it sends a message to the active call control computer to begin billing. The active call control computer then updates its call data records, and replicates the call data to the standby call control computer.

Once a call connection is established, the passenger may begin a conversation with the called party. Assume that sometime during the conversation, HAD **32** running on the Standby Call Computer, or Mon **50** running on VRU **16**, Switching Matrix **12** or Database Computer **18**, detects a hardware or software failure on the active call control computer. The HAD **32** or Mon **50** sends a Vote-To-Switch message to the standby HAD. If the standby HAD receives two such messages within a predetermined period of time, the standby HAD sends a message to the active HAD telling the active HAD to enter standby mode. The standby HAD then places the standby call control computer in active mode. The standby call control computer then retrieves the call data for this particular call from its call data record. The newly activated call control computer then sends a message to Switching Matrix **12** to send it all future data for this

particular call. Since the standby call control computer has an updated call data record, the passenger and called party can continue their conversation without any interruptions.

Had Module **32**

HAD module **32** (hereinafter referred to simply as "HAD") runs on a call control computer as a message-driven state transition engine designed to coordinate the call processing states of a Resource Manager (REM) module running on the active call control computer with a Remote Access Dip (RAD) module running on VRUs **16**. HAD talks to a MON-CV module running on VRUs **16** to exchange information about platform startup/shutdown. MON-CV in turn relays messages to RAD.

The HAD message types listed here are named generically for simplicity when used in describing the HAD activities and state transitions that follow. HAD receives the following messages:

- ImAlive—from MONs, HAD-Mate and critical processes when they have initialized; from critical processes as a heartbeat answer
- ImDead—from MONs, Had-Mate and critical processes when they are going down gracefully
- MonState—from the MONs as a heartbeat answer and report of current activity level
- RemState—from the critical process REM to report its current call processing level
- HadState—from HAD-Mate as a heartbeat answer and report of current activity level
- StateQuery—from MONs and HAD-Mate to request a heartbeat response in the form of a report on current level of activity
- GoActive—from UI for on-demand activation; from HAD during a switch-over
- GoStandby—from UI for on-demand de-activation; from HAD during a switch-over
- VoteToSwitch—from a MON to HAD-Stand when MON has detected missed responses heartbeats by HAD-CurrActy
- ImController—from its HAD-Mate when HAD-Mate going active

HAD sends the following messages:

- ImAlive—to MONs, HAD-Mate and critical processes when HAD has initialized
- ImDead—to MONs, HAD-Mate and critical processes when HAD is going down gracefully
- AliveQuery—to critical processes to request a heartbeat response in the form of an ImAlive report
- StateQuery—to HAD-Mate and MONs to request a heartbeat response in the form of a report on current level of activity
- HadState—to HAD-Mate and MONs as a heartbeat answer and report of current activity level
- GoActive—to HAD-Stand from HAD-CurrActy to bring the standby HAD up to fully active call processing level when HAD-CurrActy has lost a critical process; to MON-Cvs to bring them to fully active state; to REM to bring it to fully active state
- ImController—to its HAD-Mate and MONs when going active

Upon initialization, HAD goes through a setup process that includes reading its parameter files. One of the parameters is the default active All Server designation, DEF_ACTIVE_CS. If HAD sees that its call control computer is the default active call control computer, then HAD records itself as

being the default active HAD, or HAD-DefActy. It knows then to try to go active automatically without receiving a manual GoActive command from the User Interface (UI). The rules are:

- If HAD-DefActy sees that its HAD-Mate is not active (out-of-service or on warm standby), then it will go fully active.
- If HAD-DefActy sees that its HAD-Mate is going active or already active, then it will only go to warm standby state.

If a HAD sees that it is NOT the default active, then it also knows to only go to the warm standby state. The HAD that is not default active may be brought active by explicit command from the UI during manual switch-over or by a GoActive command from its HAD-Mate or through the automatic VoteToSwitch switch over scenario.

If either HAD process dies and re-spawns, it goes through the same initialization as it would during a cold start. The re-spawned HAD reads the DEF_ACTIVE_CS parameter and proceeds as above.

To initiate tracking of critical processes by HAD, all critical processes on the call control computer send an ImAlive report on start-up to HAD. HAD then creates a process record that contains updatable information about the process' communication status or, if appropriate, its process state. When the other monitors, MON-Op, MON-CV and HAD-Mate wake up, they also report ImAlive to HAD.

HAD uses an internal alarm routine to regularly heartbeat critical call control computer processes. HAD sends AliveQueries as heartbeats to all its critical processes every PROC_HB_INTERVAL number of seconds. All recipients of HAD's AliveQuery should respond with an ImAlive. Upon receipt of an ImAlive response from a critical process, HAD updates that process's heartbeat record. HAD keeps track of unanswered AliveQueries. If a process fails to respond to PROC_HB_MISSES number of HAD's AliveQueries, HAD may raise alarms or undergo a state transition. The PROC_HB_INTERVAL and PROC—HB_MISSES parameters are tuneable.

The UI maintains HAD's list of critical processes in a parameter file. HAD reads these parameters on startup or when UI sends it a RereadParms command.

HAD uses an internal alarm routine to regularly heartbeat the remote MON server. HAD sends StateQueries as heartbeats to the MONs, which include MON-CV and MON-Op. However, only the MON-Cv's current level of activity is important for HAD's monitoring of platform health and call processing ability. (MON-Op operates independently of the rest of the platform.) HAD sends these heartbeats every MON_HB_INTERVAL number of seconds. MONs should answer StateQueries with a MonState report. Upon receipt of a MonState report, HAD updates the MON's heartbeat and state record. HAD keeps track of unanswered StateQueries. If a MON fails to respond to MON_HB_MISSES number of HAD's StateQueries, HAD raises an alarm but does not undergo any state transition itself. Both MON_HB_INTERVAL and MON_$_{HB}$_MISSES are tuneable parameters.

a MON's state may be:

MON_OOS ("OOS" stands for "Out-of-Service")
MON_STANDBY
MON_WAIT_RAD_ACTIVE
MON_ACTIVE
MON_WAIT_RAD_OOS
MON_MAINT_$_{STANDBY}$

HAD, whether ACTIVE or STANDBY, tracks its HAD-Mate with StateQuery heartbeats. The HAD-Mate should

respond with a HadState report. Upon receipt of a HadState report, HAD updates HAD-Mate's heartbeat and state record. HAD keeps track of unanswered StateQueries. If the HAD-CurrActy fails to respond to HAD_HB_MISSES number of HAD-Stand's StateQueries, then HAD-Stand begins to look for confirmation of HAD-CurrActy communication problems in the form of a VoteToSwitch notification from any MON that has also detected communication failure with HAD-CurrActy. If HAD-Stand gets this confirming VoteToSwitch from MON within SWITCH_INTERVAL seconds of the time HAD-Stand first detected missed heartbeats, it initiates automatic quick switch-over of the platform and brings its call control computer to full activation. If HAD-Stand gets a VoteToSwitch from a MON before HAD-Stand itself had detected missed HAD-CurrActy heartbeats, then HAD-Stand starts counting the SWITCH_INTERVAL and waits for another VoteToSwitch from another MON before beginning automatic quick switch-over. If HAD-Stand switches-over and becomes the currently active HAD, it sends the other monitors an ImController announcement so that call flow may be re-directed to the new active call control computer. The parameters SWITCH_INTERVAL and HAD_HB_MISSES are tuneable. Please note that automatic quick switch-over occurs without any reference to which call control computer is designated the default active.

HAD's state may be:

HAD_OOS

HAD_STANDBY

HAD_WAIT_MONS

HAD_WAIT_REM

HAD_ACTIVE

A tracked critical process's or MON's or HAD-Mate's heartbeat status may be:

ALIVE—if process c continues to answer AliveQueries from HAD.

MISSES_HTBT—if process has failed to respond to one or more successive AliveQueries up to PROC_HB_MISSES number.

NOT_RESPONDING—if process has failed to respond to PROC_HB_MISSES successive AliveQueries, but the process is not found to be dead using kill(O).

DEAD—if a non-responding process is found to be dead using kill(O). Also, an ImDead from a critical process will cause an immediate transition to this heartbeat status, or lack thereof.

HAD relies on the following variables:

DEF_ACTIVE_CS—This is the machine name of the designated default active call control computer. It may be changed at any time. It is used by a HAD when initializing or when trying to resolve conflicting activity levels with its HAD-Mate.

PROC_HB_INTERVAL—This is the interval in seconds between AliveQuery heartbeats sent by HAD to its critical processes and StateQuery hearts to its HAD-MATE. The default is 1 second.

PROC_HB_MISSES—This is the number of successively missed responses to AliveQueries that HAD allows a critical process before declaring it NOT_RESPONDING. The default is 2.

REM_INIT_TIMER—This is the number of successively missed responses to AliveQueries that HAD-CurrActy allows REM when HAD-CurrActy has just transitioned into HAD-Wait_REM state and send REM a GoActive command.

MON_HB_INTERVAL—This is the interval in seconds between StateQuery heartbeats sent by HAD to MONs. The default is 10 seconds.

MON_HB_MISSES—This is the number of successively missed responses to StateQueries that HAD allows a MON before declaring it NOT_RESPONDING. The default is 2.

SWITCH_INTERVAL—This is the interval in seconds after the HAD-Stand has detected or received notification of HAD-CurrActy's non-responsiveness from a MON. HAD-Stand must receive a confirmation of the HAD-CurrActy problem from another MON in order to begin an automatic quick switch-over. The default is 5 seconds.

The following comprises a description of HAD states and transitions.

HAD_OOS:

When HAD initializes its starts in the out-of-service state, HAD-OOS. When all its call control computer Critical processes have initialized and sent ImAlive declarations, HAD transitions to the next standby state.

HAD_STANDBY:

In state HAD_STANDBY, all critical call control computer processes are responding to AliveQueries and HAD is talking to the remote MONs on all the CRIS units and to its HAD-Mate. HAD is considered to be on warm standby. If HAD recognizes itself to be the default active HAD, it transitions to the next waiting state if its HAD-Mate is not currently active. If HAD is told to GoActive by UI or its HAD-Mate, it transitions to the next waiting state regardless of who is the default active HAD. And if HAD receives a VoteToSwitch notification and confirmation, it transitions to the next waiting state to begin an automatic quick switch-over. If HAD is able to GoActive, it notifies everyone with an ImController announcement.

HAD_WAIT_MONS:

In this state, HAD is considered to be "going active". If HAD has arrived at this state during an automatic initialization or during a quick switch-over, then HAD looks for messages from all MON-Cvs that acknowledge HAD as active and at least one MON-CV as in standby state. When these have been received, HAD transitions to the next state and sends REM a RemGoActive command. If HAD has arrived at the HAD_WAIT_MONS state during an on-demand graceful switch-over, HAD tells the MON-Cvs to step down to standby mode. Then HAD looks for messages from all MON-Cvs that acknowledge HAD as active and at least one MON-CV as in standby state. When these have been received, HAD transitions to the next state and sends a RemGoActivef command to its critical process REM.

HAD_WAIT_REM:

In this state, HAD is still active. HAD is waiting for REM to answer the RemGoActive command with a RemGoneActive report. When HAD gets this report, it transitions to the fully active HAD_ACTIVE state and sends the MON-Cvs a GoActive command. (The MON-Cvs may already be in active state if a quick switch-over is under way.)

HAD_ACTIVE

In this state, the RAD is processing calls and interacting with REM on the call control computer. HAD is sending periodic AliveQuery and StateQuery heartbeats, updating communication and state

records and keeping track of missed heartbeat responses. If HAD sees that a critical process is not responding to its AliveQueries, or if a critical process sends an ImDead report, HAD will transition down to state HAD_OOS and send a Go-Active command to its HAD-Mate to takeover. It brings down call processing on its call control computer by sending REM a GoStandby command.

The following Tables 1 through 5 provide detailed information regarding the functions of HAD in response to certain conditions.

TABLE 1

HAD STATE: HAD_OOS

Stimulus	New State	Action
from REM or critical processes:		
ImAlive from a critical process making all critical processes alive or periodic check of received AliveQuery responses shows all critical processes alive 1-1	HAD-STANDBY	Send Mate HadGoneStandby Send Mon-Op HadGoneStandby
ImDead from a critical process or periodic check of missed AliveQuery responses exceeds threshold for critical process	*	Had already HAD_OOS.
RemGoneActive	*	Update Rem state. Alarm "Rem went Active unexpectedly". Tell RemGoStandby.
RemGoneStandby	*	Update Rem state.
from HAD-Mate or UI:		
GoActive	*	Alarm "Had cannot go ACTIVE from OOS
GoStandby	*	Alarm "Had cannot go STANDBY from OOS
from MON:		
Monstate	*	Update Mon's state.
ImAlive	*	Update Mon's status,
ImDead from Mon-CV shows all Mon-Cvs dead or periodic check of missed StateQuery responses shows all Mon-Cvs dead	*	Update Mon's state. Alarm "All CRIS are dead".
VoteToSwitch and confirmation within Switch_Interval	*	Alarm "Had-Mate dying. This Had OOS and cannot take over!".
from HAD-MATE:		
ImController	*	Ignore.
HadState	*	Update Had's state.
ImDead	*	Update Had's state. Alarm "Had-Mate dying. This Had OOS and cannot take over!".

TABLE 2

HAD STATE: HAD_STANDBY

Stimulus	New State	Action
from REM or critical processes:		
ImAlive from a critical process making all critical processes alive or periodic check of received AliveQuery responses shows all critical processes alive	*	Had already HAD_STANDBY.
ImDead from a critical process or periodic check of missed AliveQuery response exceeds threshold for critical process	HAD_OOS	Send Had-Mate HadGoneOos. Send Mons HadGoneOos. Tell Had-Mate GoActive. Alarm "This Had lost critical process. Going OOS Had-Mate taking over".
RemGoneActive	*	Update Rem state. Alarm "Rem went Active unexpectedly". Tell RemGoStandby.
RemGoneStandby	*	Update Rem status.
from HAD-Mate or UI:		
GoActive (quick)	HAD_WAIT_MONS	Log "This Had STANDBY and beginning quick switch-over!" Record Had-CurrActy = me. Tell Had-Mate GoStandby. Tell Mon ImController. Tell Had-Mate HadGone WaitMons. Send Mons MonGoneWaitMons.
GoActive (graceful)	HAD_WAIT_MONS	Log "This Had STANDBY and beginning Graceful switch-over!" Record Had-CurrActy = me. Tell Had-Mate GoStandby. Tell Mon-Cvs GoStandby. Tell Mon ImController. (*) Tell Had-Mate ImController. Send Had-Mate HadGoneWait Mons. Send Mon-Op MonGoneWaitMons.
GoStandby	*	Ignore. Had already HAD_STANDBY.
from MON:		
MonState	*	Update Mon's state
ImAlive	*	Update Mon's status
ImDead from Mon-CV making all Mon-Cvs dead or periodic check of missed StateQuery responses exceeds threshold for Mon-Cvs	*	Alarm "All CRIS are dead".
VoteToSwitch and confirmation within Switch_Interval	HAD_WAIT_MONS	Alarm "Had-Mate dying. This Had STANDBY and beginning quick switch-over!" Record Had-CurrActy = me. Tell Had-Mate GoStandby. Tell Had-Mate ImController Tell Mon ImController.

TABLE 2-continued

HAD STATE: HAD_STANDBY

Stimulus	New State	Action
		Send Had-Mate HadGoneWaitMons Send Mons HadGoneWaitMons.
initialization determination only done in HAD_STANDBY state:		
Im Had-DefActy and Had-Mate is NOT currently Active, so I will GoActive automatically.	HAD_WAIT_MONS	Log "This Had Standby and beginning automatic initialization!" Record Had-DefActy = me. Record Had-CurrActy = me. Tell Had-Mate GoStandby. Tell Mon-Cvs GoStandby. Tell Had-Mate ImController. Tell Mon ImController. Send Had-Mate HadGoneWaitMonsStandby. Send Mons HadGoneWaitMonsStandby.
I AM Had-DefActy but Had-Mate is currently Active, so I will stay Standby.	*	Record Had-DefActy = me. Record Had-CurrActy = mate
I am NOT Had-DefActy.	*	Record Had-DefActy = mate.
from HAD-Mate:		
ImController from Had-Mate but I am Had-DefActy Looks like Had-Mate go there first!	*	Record Had-DefActy = me. Record Had-CurrActy = mate.
ImController from Had-Mate and I am not Had-DefActy Looks like Had-Mate doing an automatic startup.	*	Record Had-DefActy mate. Record Had-CurrActy = mate.
HadState	*	Update Had's state.
ImDead	HAD_WAIT MONS	Alarm "Had-Mate dying. This Had STANDBY and beginning quick switch-over!" Record Had-CurrActy = me. Tell Had-Mate GoStandby. Tell Had-Mate ImController. Tell Mon ImController. Send Had-Mate HadGoneWaitMons. Send Mons HadGoneWaitMons.

TABLE 3

HAD STATE: HAD_WAIT_MONS

Stimulus	New State	Action
from REM or critical processes:		
ImDead from a critical process or periodic check of missed Alive Query responses exceeds threshold	HAD_OOS	Send Had-Mate HadGoneOos. Send Mons HadGoneOos. Tell Had-Mate GoActive. Tell RemGoStandby Alarm "This Had lost critical process. Going OOS. Had-Mate taking over".

TABLE 3-continued

HAD STATE: HAD_WAIT_MONS

Stimulus	New State	Action
for critical process		
RemGoneActive	*	Update Rem state
RemGoneStandby	*	Update Rem state
from HAD-Mate or UI:		
GoActive	*	Ignore. Had already going ACTIVE.
GoStandby	HAD_STANDBY	Send Had-Mate HadGoneStandby. Send Mons HadGone Standby. Tell RemGoStandby.
from MON:		
MonState	*	Update Mon's state
message from a Mon-CV confirms all Mon-Cvs know I am Had-CurrActy and at least one Mon-CV standby or periodic check of Mon-CV records confirms same	HAD_WAIT_REM	Send Had-Mate HadGoneWaitRem Send Mon-Op HadGoneWait Rem Tell RemGoActive.
ImAlive	*	Update Mon's status
ImDead from Mon-CV making all Mon-Cvs dead or periodic check of Mon-CV records shows all Mon-CVs dead	*	Alarm "All CRIS are dead".
VoteToSwitch and confirmation within Switch_Interval	*	Ignore. Had already going ACTIVE.
from HAD-Mate		
ImController from Had-Mate but I am Had-DefActy and also going active! This shouldn't happen, but if it does I will tell my confused Had-Mate who's boss	*	CONFLICT! Had-DefActy wins and reasserts control Tell Had-Mate GoStandby Tell Had-Mate ImController. Tell Mon ImController. Alarm "Possible Had conflict over who is active. Stay tuned"!
ImController form Had-Mate when I am going active but I am NOT Had-DefActy Now I'm the one who is confused! I must step down.	HAD_STANDBY	CONFLICT Had-DefActy wins. I am NOT Had-DefActy, so I give up! Had-Mate will probably be sending me a GoStandby anyway. This is the previous conflict reversed. Tell RemGoStandby Alarm "Possible Had conflict over who is active. Stay tuned"!
HadGoneOos or HadGoneStandby	*	Update Had's state
HadGoneWaitMons Standby or HadGoneWaitRemActive or HadGoneActive	*	Alarm "Conflicting Had states. This Had WAIT_MONS. Had-Mate is Active."
ImDead	*	Alarm "Had-Mate dying This Had already WAIT-MONS-STANDBY and going ACTIVE!"

TABLE 4

HAD STATE: HAD_WAIT_REM

Stimulus	New State	Action
from REM or critical processes;		
ImDead from a critical process or periodic check of missed Alive Query responses exceeds threshold for critical process	HAD_OOS	Send Had-Mate HadGoneOos. Send Mons HadGoneOos. Tell Had-Mate GoActive. Tell RemGoStandby. Alarm "This Had lost critical process. Going OOS. Had-Mate Had taking-over".
RemGoneActive	HAD_ACTIVE	Update Rem state Tell MonGoActive Send Mate HadGoneActive Send Mons HadGoneActive
RemGoneStandby	*	Update Rem state Alarm "Rem goneSTANDBYunexpectedly"
from HAD-Mate or UI:		
GoActive	*	Ignore. Had already going ACTIVE.
GoStandby	HAD_STANDBY	Send Had-Mate HadGoneStandby. Send Mons HadGoneStandby. Tell RemGoStandby.
from MON:		
MonState	*	Update Mon's state
ImAlive	*	Update Mon's status
ImDead from Mon-CV making all Mon-Cvs dead or periodic check of missed StateQuery responses shows all Mon-Cvs dead	*	Update Mon's state Alarm "All CRIS are dead".
VoteToSwitch and confirming vote within Switch_Interval	*	Ignore. Had already gone ACTIVE
from HAD-Mate		
ImController from Had-Mate but I am Had-DefActy and also going active! This shouldn't happen, but if it does I will tell my confused Had-Mate who's boss.	*	CONFLICT! Had-DefActy wins and reasserts control. Tell Had-Mate GoStandby Tell Had-Mate ImController. Tell Mon ImController. Alarm "Possible Head conflict over who is active. Stay tuned!"
ImController from Had-Mate when I am going active but I am NOT Had-DefActy Now I'm the one who is confused! I must step down.	HAD_STANDBY	CONFLICT! Had-DefActy wins. I am NOT Had-DefActy, so I give up! Had-Mate will probably be sending me a GoStandby anyway. This is the previous conflict reversed. Tell RemGoStandby Alarm "Possible Had conflict over who is active. Stay tuned!"
HadGoneOos or HadGoneStandby	*	Update Had-mate's state.
HadGoneWaitMons Standby or HadGoneWaitRemActive ort HadGoneActive	*	Update Had-Mate's state Alarm "Conflicting Had states. This Had is WAIT_REM. Had-Mate is ACTIVE."

TABLE 4-continued

HAD STATE: HAD_WAIT_REM

Stimulus	New State	Action
ImDead	*	Alarm "Had-Mate dying. This Had already WAIT_REM and going ACTIVE!"

TABLE 5

HAD STATE: HAD_ACTIVE

Stimulus	New State	Action
from REM or critical processes:		
ImDead from a critical process or periodic check of missed AliveQuery responses exceeds threshold for critical process	HAD_OOS	Send Had-Mate HadGoneOos Send Mons HadGoneOos. Tell Had-Mate GoActive. Tell RemGoStandby. Alarm "This Had lost critical process. Going OOS. Had-Mate taking over".
RemGoneActive	*	Update Rem state
RemGoneStandby	*	Update Rem state Alarm "Rem gone STANDBY unexpectedly". Tell RemGoActive.
from HAD-Mate or UI:		
GoActive	*	Ignore. Had already ACTIVE.
GoStandby	HAD_STANDBY	Send Had-Mate HadGoneStandby. Send Mon-Op HadGOneStandby. Tell RemGoStandby.
from MON:		
MonState	*	Update Mon's state
ImAlive	*	Update Mon's state
ImDead from Mon-CV making all Mon-Cvs dead or periodic check of missed StateQuery responses shows all Mon-Cvs dead	*	Update Mon's state Alarm "All CRIS are dead".
VoteToSwitch and confirming vote with Switch_Interval	*	Ignore. Had already ACTIVE.
from HAD-Mate		
ImController from Had-Mate and I am already fully active. This shouldn't be happening. But since ImController message is sent by Hads at the BEGINNING of going active, I think I win here.	*	Tell Had-Mate GoStandby Tell Had-Mate ImController. Tell Mon ImController. Alarm "Possible Had conflict over who is active. Stay tuned!"
HadGoneOos or HadGone Standby	*	Update Had's state
HadGoneWaitMonsStand by or	*	Update Had's state Alarm "Conflicting Had

TABLE 5-continued

HAD STATE: HAD_ACTIVE

Stimulus	New State	Action
HadGoneWaitRemActive or HadGoneActive		states. Had-Mate is ACTIVE."
ImDead	*	Alarm "Had-Mate dying. This Had already ACTIVE."

Mon Module **50**

Mon module **50** may be optimized to run on different network devices, e.g., a VRU or database computer. This detailed description will provide an outline of two types of MON modules, the first designed for a VRU (MON-CV) and the second designed for a ground to air server (GTAS) (MON-OP) used in one embodiment of the invention.

MON-CV

MON-CV runs on VRUs **16** as a message driven state transition engine designed to monitor and coordinate the call processing states of the RAD module running on VRUs **16** with the call processing states of the REM module running on the active call control computer. Mon-CV talks to the HAD-CurrActy server to receive platform startup/shutdown commands and other updates from it. MON-CV also keeps an eye on the HAD-CurrActy's communication status. If MON-CV detects any problems talking to HAD-CurrActy, it will immediately notify the standby HAD to be alert for a possible switch-over.

The MON-CV message types listed here are named generically for simplicity when used in describing the MON-CV activities and state transitions that follow. MON-CV receives the following messages:

ImAlive—from both HADS and from its critical process RAD when they have initialized and from RAD regularly as a heartbeat answer to confirm viability.

ImDead—from both HADs or RAD when they are going down gracefully.

StateQuery—from HADs to request a heartbeat response in the form of a state report.

HadQuery—from HADs as a heartbeat answer in the form of a state report.

RadGoneOos—from RAD to report its current call processing level as out-of-service.

RadGoneActive—from RAD to report its current call processing level as active.

RadGoneMoos—from RAD to report itself in a maintenance state.

GoActive—from HAD-CurrActy for normal activation, from UI for manual activation.

GoStandby—from UI or HAD for de-activation.

ImController—from the HAD going active during initialization or switch-over.

MON-CV sends the following messages;

AliveQuery—to RAD to request a heartbeat response indicating viability.

StateQuery—to HADs to request a heartbeat response in the form of a state report.

MonState—to HADs as a heartbeat answer and report of current activity level.

GoOos—to RAD to step it down from active call processing to out-of-service.

GoActive—to RAD to bring it up to fully active call processing level.

VoteToSwitch—to HAD-Stand when MON-CV has detected a communication problem with the currently active HAD.

MON-CV uses an internal alarm routine to regularly send an Alive Query heartbeat every PROC_HB_INTERVAL number of seconds. RAD should respond with an ImAlive report. Upon receipt of an ImAlive from RAD, MON-CV updates the communication status of RAD. If RAD fails to respond to PROC_HB_MISSES number of AliveQueries, MON-CV raises an alarm and transitions to an out-of-service state. The PROC_HB_INTERVAL and PROC_HB_MISSES parameters are tuneable.

To track communication status of HADs, MON-CV uses an internal alarm to regularly send them StateQuery heartbeats every HAD_HB_INTERVAL number of seconds. HADs should respond with HadState reports. Upon receipt of a HadState report from HAD, MON-CV updates the communication status of the HAD. The HAD_HB_INTERVAL parameter is tuneable.

MON-CV keeps track of missed StateQuery response. If the currently active HAD fails to respond to HAD_HB_MISSES number of MON-CV's StateQueries, MON-CV immediately notifies the standby HAD to be alert for a possible quick switch-over by sending it a VoteToSwitch notification. If within SWITCH_INTERVAL number of seconds, the voting MON-CV has not detected renewed heartbeats from the non_responding HAD-CurrActy or has not received notice from HAD-Stand of a switch-over underway, then the voting MON-CV will step itself down to a standby status because it cannot inform its RAD critical process of where to direct call flow. The HAD_HB_MISSES and SWITCH INTERVAL parameters are tuneable.

Heartbeat status for a tracked critical process or tracked HAD may be:

ALIVE—if process[HAD] continues to answer AliveQueries [StateQueries] from MON-CV.

MISSED_HTBT—if process[HAD] has failed to respond to one or more successive AliveQueries [StateQueries] up to PROC_HB_MISSES] [HAD_HB_MISSES] number.

NOT_RESPONDING—if process[HAD] has failed to respond to PROC_HB_MISSES [HAD_HB_MISSES] successive AliveQueries [StateQueries], but process is not found to be dead using kill(O).

DEAD—if a non-responding process{HAD] is found to be dead using kill(O). Also, an ImDead report from process{HAD] will cause an immediate transition to this heartbeat status.

MON-CV relies on the following parameters:

HAD_HB_INTERVAL—This is the interval in seconds between StateQuery heartbeats sent by MON-CV to the HADs. The default is 1 second.

HAD_HB_MISSES—This is the number of successive missed responses to StateQueries that MON-CV allows a HAD before declaring it NOT_RESPONDING. The default is 2.

PROC_HB_INTERVAL—This is the interval in seconds between AliveQuery heartbeats sent by MON-CV to RAD. The default is 1 second.

PROC_HB_MISSES—This is the number of successive missed responses to AliveQueries that MON-CV allows RAD before declaring it NOT_RESPONDING. The default is 2.

SWITCH_INTERVAL—This is the number of seconds that MON-CV waits for a switch-over notification or

detection of renewed HAD-CurrActy heartbeats following MON-CV's sending of a VoteToSwitch notification to HAD-Stand. If SWITCH_INTERVAL expires with no notifications, the MON-CV steps itself down to standby state.

The following provides a description of MON-CV states and transitions.

MON_OOS:

When MON-CV initializes, it starts in the out-of-service state, MON_OOS. When RAD has initialized and sent an ImAlive declaration, MON-CV transitions to the next standby state.

MON_STANDBY:

In state MON-STANDBY, MON-CV knows that RAD is alive, but RAD's circuits are still OOS. MON-CV is talking to the remote HADs on the call control computers and to RAD. When MON-CV receives an ImController announcement from the HAD going active, MON-CV records the identity so that it may tell RAD which is the current active call control computer. When HAD-CurrActy completes activation and sends MON-CV a GoActive command, MON-CV transitions to the next state and sends RAD a RadGoActive command.

MON_WAIT_RAD_ACTIVE:

In this state, MON-CV is waiting for RAD to answer the RadGoActive command with a RadGoneActive report. When RAD sends this report, the MON-CV transitions to the fully active MON_ACTIVE state.

MON_ACTIVE:

In this state, RAD is processing calls on the CRIS unit and interacting with REM on the call control computer. MON-CV is sending periodic AliveQuery heartbeats to RAD and StateQueries to both HADS, updating communication records and keeping track of missed heartbeat responses. If MON-CV sees that the currently active HAD is not responding to StateQueries, MON-CV will immediately notify the standby HAD (with a VoteToSwitch notification) to be alert for a possible quick switch-over as described above. If such a quick switch-over occurs, the newly activated HAD sends MON-CV an ImController announcement. MON-CV records which is the new HAD-CurrActy and notifies RAD with no change to RAD's current state. At any time in MON_ACTIVE state, if MON-CV sees that RAD is not responding to its AliveQueries, MON-CV will transition down to state MON_OOS and notify the HADS.

MON_WAIT_RAD_OOS:

If MON-CV must step down from active call processing due to receipt of a GoStandby command from UI or HAD, MON-CV first sends RAD a RadGoOos command. When RAD answers with a RadGoneOos report, MON-CV transitions to state MON_STANDBY.

MON_MAINT_STANDBY:

If RAD sends MON-CV a RadGoneMoos report indicating its need for maintenance, MON-CV transitions to MON_MAINT_STANDBY. When RAD must go MOOS, it busies out all CRIS circuits as they become ideal, in anticipation of CRIS shutdown. MON-CV does not try to re-activate RAD if there is a currently active call control computer. RAD can only leave its MOOS state by manual command or re-boot. In all Mon states, MON-CV responds to RAD's ImDead report, RadGoneOos report or to PROC_HB_MISSES number of successively missed heartbeat responses by RAD by transitioning down to a MON_OOS state and sending the HADs a MonGonOos report.

The following Tables 6 to 10 provide a detailed description of the functions of MON-CV in response to certain conditions.

TABLE 6

MON-CV STATE: MON-OOS

Stimulus	New State	Action
from RAD:		
ImAlive	MON_STANDBY	Update Rad. status. Tell Had GoneStandby
RadGoneActive		Update Rad.'s state. Alarm "Rad. went ACTIVE unexpectedly"
RadGoneMoos	MON_MAINT_STANDBY	Update Rad.'s state.
from HAD:		
ImDead		Ignore. Update Had's state. Mon not yet STANDBY.
ImController		Ignore. Mon not yet STANDBY.
periodic check of missed StateQuery responses exceeds threshold for HAD-CurrActy		Ignore. Update Had's state. Mon not yet STANDBY.
GoActive		Alarm "Mon cannot go ACTIVE from OOS."
GoStandby		Alarm "Mon cannot go STANDBY from OOS."
StateQuery		Send Had MonOos.

TABLE 7

MON-CV STATE: MON_STANDBY

Stimulus	New State	Action
from RAD:		
ImDead	MON_OOS	Update Rad.'s state. Send Had MonGoneOos.
RadGoneMoos	MON_MAINT_STANDBY	Update Rad.'s state. Send Had MonGoneMainStandby.
periodic check of missed AliveQuery responses exceeds threshold for Rad.	MON_OOS	Update Rad.'s state. Send Had MonGoneOos.
from HAD:		
ImDead		Update Had's state. Mon not yet in the voting business
ImController		Record which is Had-CurrActy.
periodic check of missed StateQuery responses exceeds threshold for Had-CurrActy		Update Had's state. Mon Not yet in the voting business.
GoActive	MON_WAIT_RAD_ACTIVE	Record which is Had-CurrActy. Tell Rad. GoActive. Send Had MonGonewaitRadActive.
StateQuery		send Had MonWaitRadActive

TABLE 8

MON-CV STATE: MON_WAIT_RAD_ACTIVE

Stimulus	New State	Action
from RAD:		
ImDead	MON_OOS	Update Rad.'s state. Setid Had MonGoneOos.
RadGoneOos		Update Rad.'s state. Alarm "Rad. went OOS unexpectedly" Tell Rad. GoActive.
RadGoneActive	MON_ ACTIVE	Update Rad.'s state. Send Had MonGoneActive.
periodic check of missed AliveQuery responses exceeds threshold for Rad.	MON_OOS	Update Rad.'s state. Send Had MonGoneOos.
from HAD:		
ImDead from Had-CurrActy		Send Had-Stand VoteToSwitch and begin SWITCH_INVERVAL wait.
ImController		Record Had-CurrActy. Notify Rad.
periodic check of missed StateQuery responses exceeds threshold for Had-CurrActy		Send Had-Stand VoteToSwitch and begin SWITCH_INTERVAL wait.
GoStandby	MON_ WAIT_ RAD_OOS	Tell Rad. GoOos. Send Had MonGoneWaitRadActive.

TABLE 9

MON-CV STATE: MON_ACTIVE.

Stimulus	New State	Action
from RAD:		
ImDead	MON_OOS	Update Rad.'s state. Send Had MonGoneOos.
RadGoneOos	MON_ STANDBY	Update Rad.'s state. Alarm "Rad. went OOS unexpectedly." Send Had MonGoneStandby.
RadGoneMoos	MON_ MAINT_ STANDBY	Update Rad.'s state. Send Had MonGoneMaintStandby.
periodic check of missed AliveQuery responses exceeds threshold for Rad.	MON_OOS	Update Rad.'s state. Send Had MonGoneOos.
from HAD:		
ImDead from Had-CurrActy		Send Had-Stand VoteToSwitch and begin SWITCH_INTERVAL wait.
ImController		Record Had-CurrActy. Notify Rad.
periodic check of missed StateQuery responses exceeds threshold for Had		Send Had-Stand VoteToSwitch and begin SWITCH_INTERVAL wait.
GoStandby	MON_ WAIT_ RAD_OOS	Tell Rad. GoOos. Send Had MonGoneWaitRadOos
StateQuery		Send Had MonStandby.

TABLE 10

MON-CV STATE: MON_WAIT_RAD_OOS

Stimulus	New State	Action
from RAD:		
ImDead	MON_OOS	Update Rad.'s state. Send Had MonGoneOos.
RadGoneOos	MON-STANDBY	Update Rad.'s state. Send Had MonGoneStandby
RadGoneMoos	MON-MAINT-STANDBY	Update Rad.'s state. Send Had MonGoneMainStandby.
periodic check of missed AliveQuery response exceeds threshold for Rad.	MON_OOS	Update Rad.'s state. Send Had MonGoneOos.
from HAD.		
ImDead		Ignore. Mon going STANDBY.
ImController		Ignore. Mon going STANDBY.
periodic check of missed StateQuery responses exceeds threshold for Had-CurrActy		Ignore. Mon going STANDBY.

In all Mon states, MON-CV responds to RAD's ImDead report, RadGoneOos report or to PROC_HB_MISSES number of successively missed heartbeat responses by RAD by transitioning down to a MON OOS state and sending the HADs a MonGonOos report. In any state, MON answers HAD's StateQueries with a MonState report.

If MON-CV bounces, it re-spawns and reinitializes as though it were undergoing a cold start. MON-CV makes no assumptions about any prior states. Currently, RAD bounces are only acknowledged if MON-CV receives an ImDead report from RAD when RAD is dying gracefully. In that case, MON transitions to state MON_OOS. Once RAD has re-initialized and sent MON-CV an ImAlive report, MON will transition to MON-STANDBY and proceeds as usual.

MON-OP (GTAS)

MON-Op runs as a message-driven state transition engine on the GTAS designed to monitor and coordinate the states of the GTAS's critical processes. MON-Op also keeps a record of the current states of the active and standby HAD monitors on the call control computer. If MON-Op notices any problems in communication with the currently active HAD, it will immediately notify the standby HAD to be alert for a possible switch-over. Because the GTAS can run independently of the rest of the platform, MON-Op needs to have only two states, out-of-service or active.

The Mon-Op message types listed here are named generically for simplicity when used in describing the MON-Op activities and state transitions that follow. Mon-Op receives the following messages:

ImAlive—from both HADs and from GTAS critical processes when they have initialized and from critical processes regularly as a heartbeat to confirm their viability.

ImDead—from both HADs or critical processes when they are going down gracefully.

StateQuery—from both HADs to request a heartbeat response in the form of a state report.

HadState—from both HADs as a heartbeat response and report of current activity level.

ImController—from the HAD going active during initialization or switch-over.

ON-Op sends the following messages:

AliveQuery—to critical processes to request a heartbeat response indicating viability.

StateQuery—to both HADs to request a heartbeat response in the form of a state report.

MonState—to both HADs' as a heartbeat answer indicating viability.

VoteToSwitch—to HAD-Stand when MON-Op has detected a communication problem with the currently active HAD.

To keep a current record of the states of the call control computers, MON-Op uses an internal alarm routine to regularly send the HADs StateQuery heartbeats every HAD_HB_INTERVAL number of seconds. Each HAD should respond with a HadState report. If the currently active HAD fails to respond to HAD_HB_MISSES number of MON-OP's StateQueries, MON-Op immediately notifies the standby HAD to be alert for a possible switch-over by sending it a VoteToSwitch notification. The HAD_HB_INTERVAL and HAD_HB_MISSES parameters are tuneable.

It is worthy to note that because the GTAS runs independently of the rest of the platform, MON-OP does not need to change its state or redirect any call flow upon failure of the HAD-CurrActy, unlike the MON-CV which must take certain action within SWITCH_INTERVAL number of seconds of sending a VoteToSwitch.

HAD's state may be:

HAD_OOS

HAD_STANDBY

HAD_WAIT_MONS

HAD_WAIT_REM

HAD_ACTIVE

To track communication status of critical servers on the GTAS, MON-Op uses the internal alarm to regularly send them AliveQuery heartbeats every PROC_HB_INTERVAL number of seconds. All recipients of MON-Op's AliveQueries should respond with ImAlive reports. MON-Op keeps track of missed AliveQuery responses. If any of its critical processes fails to response to PROC_HB_MISSES number of MON-Op's AliveQueries, then MON-Op raises an alarm and transitions to an out-of-service state. Both PROC_HB_INTERVAL and PROC_HB_MISSES are tuneable parameters.

MON-OP may need to handle multiple instances of the same critical process server. Therefore, the simplifying assumption—that one live instance of an multiply-instanced critical server is "enough" for the platform to maintain active call processing—allows MON-Op to assign an aggregate status of ALIVE to any multiply-instanced process having that one live instance.

The communications status of a tracked process or HAD may have one of the following values:

ALIVE—if process[HAD] continues to answer AliveQueries [State Queries] from MON-op.

MISSED_HTBT—if process[HAD] has failed to respond to one or more successive AliveQueries [StateQueries] up to PROC_HB_MISSES [HAD_HB_MISSES] number.

NOT_RESPONDING—if process {HAD} has failed to respond to PROC_HB_MISSES [HAD_HB_MISSES] successive AliveQueries [State Queries}, but is not found to be dead using kill(O).

DEAD—if a non-responding process[HAD] is found to be dead using kill(O). Also, receipt of an ImDead message from a process will cause an immediate transition to this heartbeat status.

Mon-Op relies on the following parameters:

HAD_HB_INTERVAL—This is the interval in seconds between StateQuery heartbeats sent by MON-op to the HADs. The default is 1 second.

HAD_HB_MISSES—This is the number of successive missed responses to StateQueries that MON-op allows a HAD before declaring it NOT_RESPONDING. The default is 2.

PROC_HB_INTERVAL—This is the interval in seconds between AliveQuery heartbeats sent by MON-Op to its critical processes. The default is 1 second.

PROC_HB_MISSES—This is the number of successive missed responses to AliveQueries that MON-Op allows a critical process before declaring it NOT_RESPONDING. The default is 2.

The following comprises a general description of MON-Op states and transitions.

MON_OOS:

When MON-Op initializes, it starts in the out-of-service state, MON OOS. When each critical process has initialized and send an ImAlive declaration, MON-Op transitions to a fully active state.

MON_ACTIVE:

In this state, MON-Op knows that the GTAS is able to process calls. MON-Op is sending periodic AliveQuery heartbeats to the critical processes and StateQuery heartbeats to the HADs. If MON-Op sees that the currently active HAD is not responding to StateQueries, MON-Op immediately notifies the standby HAD with a VoteToSwitch of the need to be alert for a possible switch-over. If MON-Op sees that a critical process is not responding to its AliveQueries, MON-Op will transition down to state MON-OOS and notify the HADS.

The following Tables 11 and 12 describe in detail the functions of MON-Op in response to certain conditions.

TABLE 11

MON-Op STATE: MON_OOS

Stimulus	New State	Action
from critical processes:		
periodic check of received AliveQuery responses shows all critical processes alive	MON_ACTIVE	Send Had MonGoneActive
ImDead from last living instance of multiply-instanced critical process or from a singly-instanced critical process	*	Update proc's status Mon-Op already OOS.
periodic check of missed AliveQuery responses exceeds threshold for last living instance of multiply-instanced critical process or for a	*	Ignore. Mon-Op already OOS.

TABLE 11-continued

MON-Op STATE: MON_OOS

Stimulus	New State	Action
singly-instanced critical process from HAD:		
ImDead from Had-CurrActy	*	Ignore. Mon-Op not yet in the voting business
HadState	*	Update Had's state
ImController	*	Ignore. Mon-Op not yet ACTIVE.
periodic check of missed StateQuery responses exceeds threshold for Had-CurrActy	*	Ignore. Mon-Op not yet in the voting business

TABLE 12

MON-Op STATE: MON_ACTIVE

Stimulus	New State	Action
from critical processes:		
ImDead from last living instance of multiply-instanced critical process or from singly-instanced critical process	MON_OOS 5	Send Had MonGonOos.
periodic check of missed AliveQuery responses exceeds threshold for last living instance of multiply-instanced critical process or for singly-instanced critical process	MON_OOS	
from HAD:		
ImDead from Had-CurrActy	*	Send Had-Stand VoteToSwitch.
HadState		Update Had's state
ImController	*	Record Had-CurrActy.
periodic check of missed StateQuery responses exceeds threshold for Had-CurrActy	*	Send Had-Stand VoteToSwitch.

Although various embodiments are specifically illustrated and described herein, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention. For example, although a pair of call control computers are used for CPP **10**, it can be appreciated that any number of call control computers can be used and still fall within the scope of the invention. In addition, although the communications monitoring processes were described with reference to CPP **10**, it can be appreciated that these processes can be implemented on other network devices and still fall within the scope of the invention.

What is claimed is:

1. A method for processing call data, comprising the steps of:

replicating call data from a first server in active mode to a second server in standby mode;

monitoring said first server by said second server and other network devices for a fault condition; and

switching said first server to standby mode and said second server to active mode if a fault condition is detected.

2. The method of claim **1**, wherein said step of replicating call data comprises the steps of:

receiving the call data at said first server;

processing the call data at said first server;

updating a call data record for said first server to reflect the call data;

sending the call data to said second server; and

updating a call data record for said second server to reflect the call data.

3. The method of claim **1**, wherein said step of monitoring comprises the steps of:

querying said first server by said network devices to detect a fault condition; and

sending a message from said network devices to said second server of a detected fault condition.

4. The method of claim **3**, wherein said step of switching comprises the steps of:

receiving at said second server said messages;

determining whether said messages reaches a predetermined threshold number, and if so:

switching said second server from standby mode to active mode; and

sending a message from said second server to said first server to switch to standby mode.

5. The method of claim **4**, further comprising the step of sending a message to said network devices to redirect call data to said second server.

6. The method of claim **1**, further comprising the steps of:

receiving static call data at a database;

storing said static call data in a static call data profile at said database; and

replicating said static call data to said first and second servers if said static call data is updated.

7. The method of claim **6**, wherein said step of replicating comprises the steps of:

receiving said static call data at said first and second servers; and

updating a static call data profile for said first server, and a static call data profile for said second servers.

8. The method of claim **7**, further comprising the step of auditing said call data records and said static call data profiles on a periodic basis to ensure data synchronization.

9. A method for processing call data, comprising the steps of:

receiving the call data at a first server in an active mode;

processing the call data at said first server;

updating a call data record for said first server to reflect the call data;

replicating the call data to a second server in a standby mode;

monitoring said first server by said second server and other network devices for a fault condition; and

switching said first server to standby mode and said second server to active mode if a fault condition is detected.

10. The method of claim **9**, further comprising the steps of:

receiving the replicated call data at said second server; and

updating a call data record for said second server to reflect the replicated call data.

11. The method of claim **10**, further comprising the step of sending a message that said first server has switched to standby mode and said second server has switched to active mode.

12. An apparatus for processing calls, comprising:

a first call control computer in active mode for receiving call data;

a second call control computer in standby mode coupled to said first call control computer;

means for replicating said call data from said first call control computer to said second call control computer;

means for monitoring said first call control computer to detect failure of said first call control computer; and

means for switching said second call control computer to active mode and said first call control computer to standby mode if said failure occurs.

13. The apparatus of claim **12**, further comprising a database coupled to said first and second call control computers.

14. The apparatus of claim **13**, wherein said call information comprises static call information and dynamic call information, and said database stores said static information.

15. The apparatus of claim **14**, further comprising a means for replicating said static call information on said first and second call control computers.

16. The apparatus of claim **13**, wherein said means for replicating replicates static call information on said first and second call control computers whenever said static call information is modified.

17. The apparatus of claim **12**, wherein said means for monitoring comprises:

means for remotely monitoring said first and second call control computers; and

means for locally monitoring said first and second call control computers.

18. The apparatus of claim **17**, wherein said means for locally monitoring comprises:

means for setting said first call control computer in active mode and said second call control computer in standby mode;

means for initializing said first call control computer in active mode;

means for determining whether a set of internal processes within said first call control computer are running within normal parameters; and

means for sending a message to said second call control computer to switch from standby mode to active mode if said set of internal processes are not running within normal parameters.

19. The apparatus of claim **17**, wherein said means for remotely monitoring comprises:

means for determining whether a set of internal processes within said first call control computer are running within normal parameters; and

means for sending a message to said second call control computer voting to switch said second call control computer from standby mode to active mode if said set of internal processes are not running within normal parameters.

20. The apparatus of clam **12**, wherein said means for switching comprises:

means for receiving at said second server vote-to-switch messages;

means for determining whether said messages reaches a predetermined threshold number, and if so:

means for switching said second server from standby mode to active mode; and

means for sending a message from said second server to said first server to switch to standby mode.

21. The apparatus of claim **20**, further comprising means for sending a message to said network devices to redirect call data to said second server.

22. A computer for performing call processing, comprising:

a memory containing:

a computer program for replicating call data from a first server in active mode to a second server in standby mode;

a set of computer programs for monitoring said first server by said second server and other network devices for a fault condition;

a computer program for switching said first server to standby mode and said second server to active mode if a fault condition is detected; and

a processor for running said programs.

23. A computer-readable medium whose contents cause a computer system to perform a remote procedure call, the computer system having a computer program that when executed performs the steps of:

replicating call data from a first server in active mode to a second server in standby mode;

monitoring said first server by said second server and other network devices for a fault condition; and

switching said first server to standby mode and said second server to active mode if a fault condition is detected.

* * * * *

NCIC OPERATOR SERVICES™

NEWS & EVENTS

HOME 1
INFORMATION ABC 2
SERVICES DEF 3
International
Domestic
Payphone
Prepaid Card
PTT
NEWS & EVENTS GHI 4
EMPLOYMENT JKL 5
LINKS MNO 6
CONTACT US PRS 7

Press Releases

09/30/99	NCIC rolls out it's new payphone referral program
09/15/99	NCIC announces call center agreement with IDA-Tel
09/01/99	NCIC introduces new Prepaid Calling Card Program for wholesale and retail services
09/01/99	NCIC announces agreement with Telefonica Centro America
09/01/99	NCIC establishes co-location facilities for Vozcom International Communications
08/01/99	NCIC completes final phase of Internet Reporting Project.
07/30/99	Dot Com Technologies announces lower pricing on it's PC to phone product
05/15/99	Dot Com Technologies announces continued support of VocalTec 3.X gateways
01/01/98	Dot Com Technologies launches InternetPhoneCall.com
12/01/97	Dot Com Technologies rolls out InternetCollect.com

NCIC rolls out it's new payphone referral program **September 30, 1999**

Today, NCIC implemented it's new payphone referral program. Through this program, NCIC has strategically posted it's

referral website on various search engines, so that store owners trying to find a payphone provider will immediately find our payphone website. These location owners will then be routed to the nearest NCIC customer that provides payphone services. Initial tests on the service have been well received and we anticipate about 200 inquiries per month..

[back to top]

NCIC announces call center agreement with IDA-Tel — **September 15, 1999**

NCIC will be handling both call center services and Internet Telephony transmission services for IDA-Tel, an internet telephony provider based in Tyler, Texas. IDA-Tel is the front runner in residential Internet Telephony outside of the US. IDA-Tel provides customers with user friendly equipment that attaches to a standard telephone that allows customers to access the internet without a PC and allows them to make inexpensive phone calls via the internet. For additional information, go to http://www.ida-tel.com .

[back to top]

NCIC introduces new Prepaid Calling Card Program for wholesale and retail services — **September 1, 1999**

NCIC has installed it's new prepaid calling platform for the wholesale and retail markets. The platform is built around a Cisco VCO4K switch utilizing Arbinet Soft-Net control systems. The platform is based on the traditional telecom prepaid platform, but also ties into the DotCom internet telephony network for world-wide internet termination for economical and high quality international calling.

[back to top]

NCIC announces agreement with Telefonica Centro America — **September 1, 1999**

NCIC finalized it's contracts with Telefonica Centro America Guatemala and El Salvador. Through this agreement, NCIC will handle collect calling for the carriers' 2 Central American divisions. Telefonica's growth into Central America has been explosive and it expects to be the dominant collect calling provider within the next year.

[back to top]

NCIC establishes co-location facilities for Vozcom International Communications — **September 1, 1999**

Today, Vozcom Communications installed an internet telephony

switch on the premises of NCIC's switching center for wholesale termination and origination. As part of this partnership, NCIC will assist in equipment management and network management for the Vozcom switch.

[back to top]

NCIC completes final phase of Internet Reporting Project. **August 1, 1999**

NCIC completed it's new internet call reporting website today. Included in the call reporting are the following reports:

- Daily Call Detail Reporting
- Daily account call and revenue totals
- Daily traffic summary by account name and ANI
- Daily and weekly comparisons
- Monthly comparisons
- Strategic graphs for traffic analyses
- Commission summaries
- Live Operator Statistics
- Database manipulation

All of these reportings help in overall route management and call accounting. Future reports include non-producing ANIs and revenue manipulation.

[back to top]

Dot Com Technologies announces lower pricing on it's PC to phone product **July 30, 1999**

DotCom has recently lowered pricing on all countries for customers using Internet Prepaid PC to Phone services. As of today, the rates posted on the Internet Prepaid website are the lowest rates available for PC to Phone calling. Visit our website at http://www.internetphonecall.com .

[back to top]

Dot Com Technologies announces continued support of VocalTec 3.X gateways **May 15, 1999**

DotCom will continue to support the older versions of internet telephony gateways in order to accommodate it's partners until they have a chance to upgrade to newer versions of Vocaltec gateways.

[back to top]

Dot Com Technologies launches InternetPhoneCall.com **January 1, 1998**

Dot Com Technologies launched www.internetphonecall.com today. This is the worlds' lowest priced prepaid internet telephony site. Customers can instantly establish an account through their web browser with their credit card. Utilizing the free software, customers can call any regular telephone in the world! You can try it for yourself at www.internetphonecall.com.

[back to top]

Dot Com Technologies rolls out InternetCollect.com **December 2, 1997**

Dot Com Technologies rolled out InternetCollect.com today. This is the worlds' first internet operator service to be offered on the world wide web. Customers can instantly connect to one of our operators through their web browser. The operator can then connect them to anyone in the U.S. for only 10 cents a minute! We are looking forward to the possibilities that internet telephony has to offer us in the changing world of telecommunications. You can try it for yourself at www.internetcollect.com.

[back to top]

For more information please contact:

Network Communications International Corporation
606 E. Magrill Longview, TX 75601
Tel: (+1) 903-757-4455
FAX: (+1) 903-757-4899
Email: info@ncic.com

[HOME] [INFORMATION] [SERVICES] [NEWS & EVENTS]
[EMPLOYMENT] [LINKS] [CONTACT US]